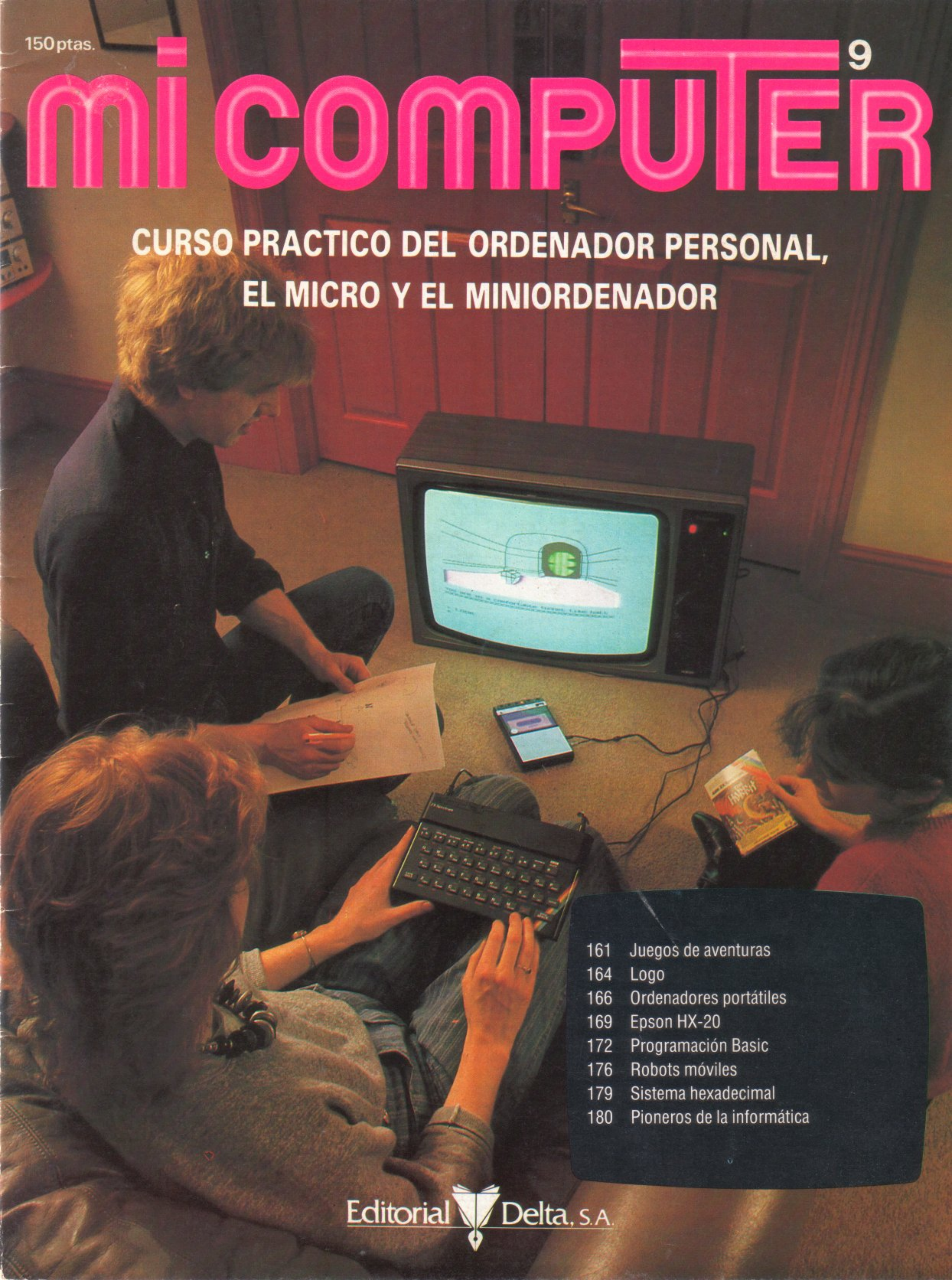

150 ptas.
mi computer
9
CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR
Editorial Delta, S.A.

MI COMPUTER CURSO PRACTICO
DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona, y comercializado en exclusiva por Distribuidora Olimpia, S.A., Barcelona

Volumen I - Fascículo 9

Director: José Mas Godayol
Director editorial: Gerardo Romero
Jefe de redacción: Pablo Parra
Coordinación editorial: Jaime Mardones
Asesor técnico: Roberto Quiroga

Redactores y colaboradores: G. Jefferson, R. Ford, S. Tarditti, A. Cuevas
Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.° - Barcelona-8
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador,* se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-85822-84-6 (tomo 1)
84-85822-82-X (obra completa)
Depósito Legal: B. 52-84

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 148403
Impreso en España - Printed in Spain - Marzo 1984

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, Madrid-34.
Distribuye para Argentina: Viscontea Distribuidora, S.C.A., La Rioja 1134/56, Buenos Aires.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio Blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Ferrenquín a Cruz de Candelaria, 178, Caracas, y todas sus sucursales en el interior del país.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 16 690 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 087 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 3371872 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Distribuidora Olimpia (Paseo de Gracia, 88, 5.°, Barcelona-8), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Distribuidora Olimpia, en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.

Editorial Delta, S.A.

Juegos de aventuras

El ordenador puede ser una fuente de diversión. En un juego de aventuras, uno no es espectador: es el protagonista

Pronuncie la palabra "aventura" y la mayoría de la gente pensará en un libro, una película, un programa de televisión e incluso, quizá, en una experiencia personal. Pero muchas personas pensarán en los ordenadores, porque existe multitud de propietarios de ordenadores para quienes la palabra Aventura (con A mayúscula) se encuentra asociada a una clase de juegos por ordenador muy específica.

Para hacernos una idea de en qué consisten las aventuras por ordenador, comparémoslas con los libros. Cuando se lee una historia de aventuras se disfruta con los peligros, los misterios y los sucesos emocionantes; pero el protagonista es otra persona. En una aventura por ordenador, en cambio, uno no es un espectador, sino que forma parte de la acción. Como principal protagonista de la historia, se encuentra inmerso en la acción y es uno mismo quien está viviendo la experiencia.

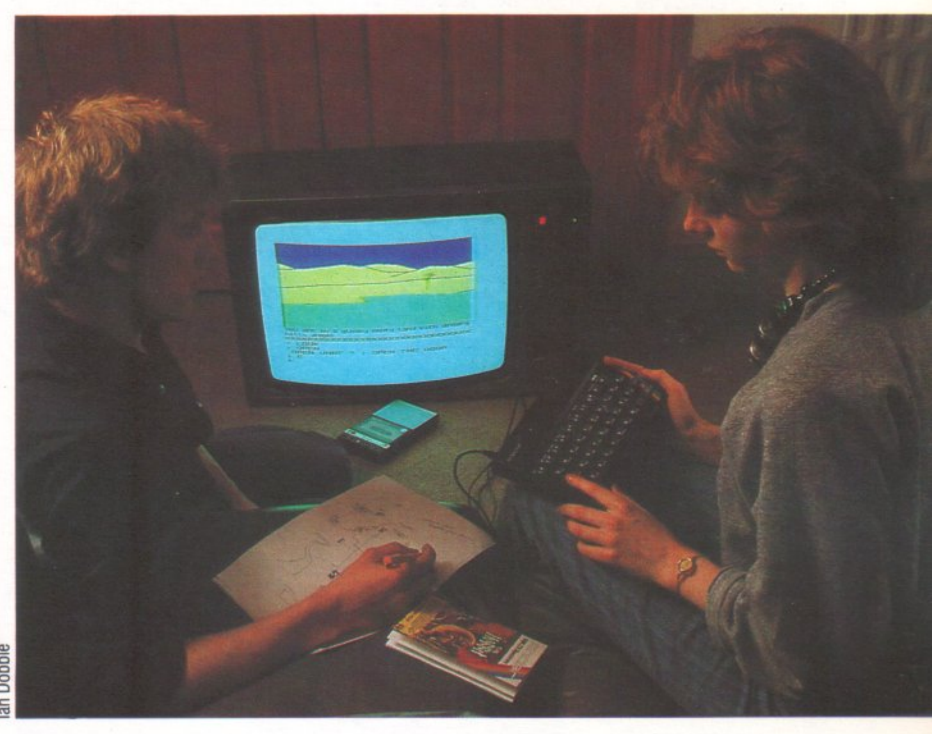
En un libro, el lector no puede modificar el curso de los acontecimientos de la narración. Los sucesos siempre son los mismos y se producen en un orden determinado, y por más que se los vuelva a leer una y otra vez, nada variará en lo más mínimo. En cambio, en una aventura por ordenador sus decisiones, pareceres y acciones determinan la forma en que se desarrolle el argumento. Puede haber cualquier número de variaciones en cuanto al orden de los acontecimientos y muchos finales diferentes, algunos agradables y otros no tanto. En un juego de aventuras lo esencial es que uno es parte activa, aunque rodeado del confort y la seguridad de su propio hogar.

Cada aventura se desarrolla en un ambiente diferente. Éste puede ser un extraño mundo subterráneo, una feria al aire libre, una ciudad fantasma, otro planeta, una tierra mítica; en realidad, cualquier lugar. La acción se puede situar en el pasado, en el presente o en el futuro.

Por lo general la aventura posee una trama coherente que incluye un objetivo final. Por ejemplo, la meta puede ser huir de un planeta desconocido, encontrar y destruir a un genio del mal, rescatar a una princesa, hallar un tesoro o esclarecer un crimen y arrestar al culpable.

Por muy interesantes que le parezcan estos ingredientes, el verdadero placer de jugar a un juego de aventuras reside en solucionar los enigmas. Éstos constituyen una parte esencial de la aventura. Se puede encontrar un enigma en una de cuatro situaciones. La situación más común consiste en tener que resolverlo para poder seguir avanzando; por ejemplo, al hallarse ante el peligro de un puente a punto de derrumbarse. En otro sentido, el enigma suele ser una pista falsa: después de haber atravesado un desfiladero con el fin de abordar a la extraña figura que lo había estado observando desde la otra orilla, usted sólo encuentra ante sí un enorme espejo. Hay enigmas cuya resolución ayuda a completar el juego exitosamente, aunque no sean un factor esencial: por ejemplo, el descubrimiento de un pasaje secreto que conduce hasta una ciudad de enanos viciosos. Puede ser una cuestión de vida o muerte: usted se encuentra solo y abandonado en el saliente de una montaña, y no existe forma alguna de seguir ascendiendo o de retroceder, y no tiene ni comida ni bebida.

Los enigmas se resuelven utilizando el sentido común y no requieren ninguna experiencia ni conocimientos especiales. No obstante, el aventurero ha de estar alerta, ya que las claves para la solución siempre se han de hallar en el texto o bien se deben deducir del mismo. En los juegos de aventuras bien escritos no existen elementos dejados al azar, excepto en dosis muy pequeñas.


Ian Dobbie

Un papel preponderante
Los juegos de aventuras, como *Dungeons and dragons* (Calabozos y dragones), existen ya desde hace muchos años; pero en la actualidad el ordenador personal los ha puesto al alcance de la mano de una gama de jugadores potenciales mucho más amplia. Los juegos de aventuras empiezan a competir, en cuanto a popularidad, con los juegos recreativos. Aunque la mayoría de ellos aún son para un solo jugador, pueden ser una absorbente fuente de recreación para todos los miembros de la familia. El jugador asume el papel de protagonista

Al jugar a una aventura probablemente encontrará objetos, mensajes y personajes que parecerán irrelevantes en el contexto de la historia. No obstante, siempre debe tener presente que en esta clase de juegos prácticamente todo tiene un fin determinado, incluso aunque, en ocasiones, esa finalidad específica sea la de despistarlo. ¿Qué significado puede tener un montón de botellas de ron rotas? ¿Cómo debe uno interpretar una voz profunda que le dice «REPITA»? ¿Qué utilidad práctica puede ofrecer un lodazal maloliente? ¿Por qué ese felpudo estaba clavado al suelo si no había nada debajo de él? Todos estos enigmas están incluidos en varios juegos de aventuras y desempeñan un papel esencial en la trama. Cuando se encuentra un objeto por primera vez, independientemente de lo común o extraño que resulte, es muy improbable que

se tenga de inmediato la certeza de su importancia; sin embargo, es seguro, como dos y dos son cuatro, que el jugador lo necesitará mucho antes de lo que cree.

Muchos juegos de aventuras poseen un pequeño laberinto en el que cada sitio o habitación se describe en idénticos términos. La única manera fiable de trazar un mapa de ruta a través de este tipo de laberintos, consiste en imitar a Hansel y Gretel, es decir, dejar caer pequeños objetos en el sendero que se sigue, con el fin de poder identificar cada cuarto. Esta estratagema se ha hecho tan conocida que algunos programadores han ideado problemas adicionales; por ejemplo, que un ladrón vaya tras el aventurero y, con la mayor discreción, cambie la disposición de los objetos que sirven de orientación.

En algunas aventuras, si bien para completar el juego exitosamente se han de resolver todos los enigmas y superar todos los escollos, el orden en que se resuelvan los misterios y se alcancen los objetivos carece de importancia. Este tipo de juegos se opone a aquellos en los cuales existe un único camino para llegar a un desenlace triunfal.

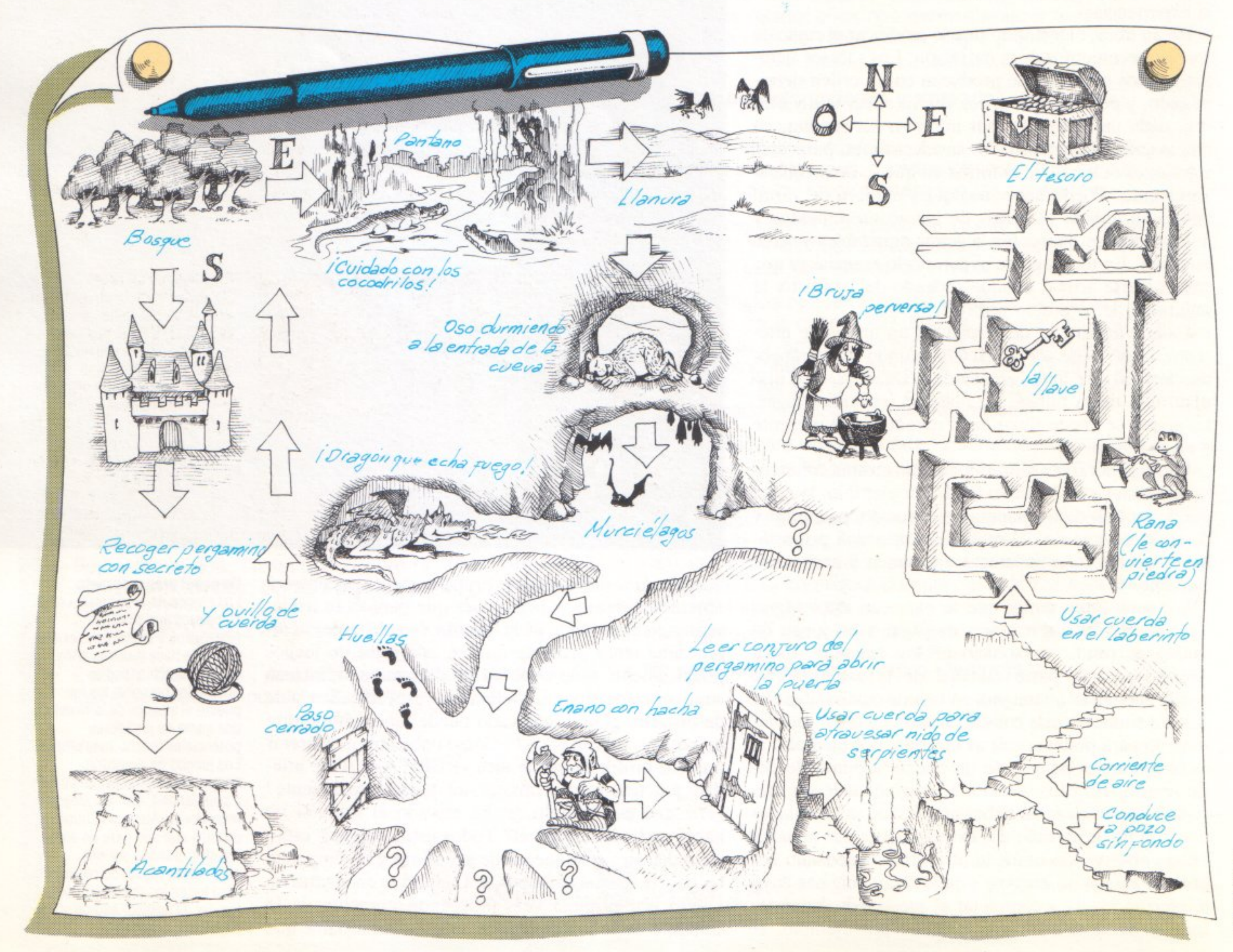
Una buena aventura puede ocuparle horas, incluso semanas de juego hasta que se descubren todos sus secretos. Entonces, cuando decida suspender el juego, estará capacitado para guardarlo en la fase en que se encuentre, en una cassette o en un disco. Este recurso también resulta útil cuando se ha llegado a una etapa peligrosa de la aventura, porque continuar avanzando impertérrito a la deriva hacia un grupo de orcas, provisto apenas de una lámpara y una botella de agua, es proceder con excesiva temeridad. Un aventurero prudente "congelará" el juego antes de tener que vérselas con las orcas. Luego, si éstas deciden terminar con los días del aventurero, por lo menos existirá la posibilidad de reanudar el juego después e intentar un curso de acción diferente, lo cual resultará mucho menos desalentador que volver a empezar la aventura desde el principio.

Incluso aunque el jugador sea muerto, ello no implica necesariamente un final definitivo. Algunos autores permiten que el difunto resucite, a menudo en medio de una humareda color naranja, a veces perdiendo puntos o algunas de sus posesiones, y siempre apareciendo en un lugar más bien indeseable, como, por ejemplo, el infierno o en medio de la nada.

¿Cómo participar y comunicarse con el programa? Éste puede dirigirse directamente al jugador o bien éste puede ser representado por un "muñeco", personaje al que se controla mediante órdenes. El ordenador actúa al mismo tiempo como intérprete de los deseos del aventurero y como narrador. El jugador da entrada a las órdenes a través del teclado y éstas se visualizan en la pantalla.

Algunas aventuras visualizan en pantalla sólo el texto, otras únicamente gráficos, el resto, una mezcla

Lectura de un mapa
En un juego de aventuras, el jugador o el personaje que éste representa se ha de mover en un vasto territorio o ambiente; es muy corriente que exista una red de pasadizos subterráneos. Algunos fabricantes incluyen pistas en los manuales de instrucción para aquellos jugadores que, llegado un momento, se queden absolutamente atascados; pero la única forma de descubrir el trazado consiste en abrirse camino a través de él. Para completar un juego de este tipo se pueden invertir semanas enteras, de manera que es imprescindible llevar un boceto esquemático que le indique en qué lugares ha estado y los obstáculos que ha encontrado a su paso



David Higham

de ambos. Los efectos sonoros se utilizan por lo general en los juegos de aventuras exclusivamente con gráficos. Las aventuras de texto se pueden comparar con un libro sin ilustraciones, en el cual los lugares, objetos y acontecimientos se describen mediante palabras. En la mayor parte de los juegos que combinan texto y gráficos, éstos sirven de complemento a las descripciones del texto y a menudo son escenas estáticas de objetos y lugares. Estas escenas pueden ser desde unos sencillos trazados lineales hasta imágenes muy detalladas. En las aventuras con gráficos, éstos normalmente se utilizan en mapas estilizados o imágenes del terreno, así como para representar los interiores de las construcciones. Los personajes y los objetos se representan mediante símbolos o figuras en miniatura. En este caso, la acción del jugador por lo general queda limitada a una pequeña cantidad de órdenes, básicamente pulsaciones individuales de teclas para hacer mover y controlar al personaje.

La visualización de texto en una aventura comprende, por lo común, tres elementos: dónde se halla el jugador, qué puede ver y a dónde puede ir. Por ejemplo, el texto en pantalla puede decir: «Usted se encuentra en la espesura de un bosque. Sobre su cabeza, el cielo no es más que una serie de manchas entre el denso follaje. Un sendero muy transitado lleva hacia el este y hacia el oeste. Delante, hacia el norte, se abre un abismo. En el suelo está clavada una espada; enroscada en ella hay una serpiente verde». Al jugador se le describen los alrededores, se le dan algunas de las direcciones que puede tomar y se le indican los objetos que están a la vista.

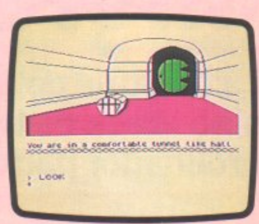
Las órdenes constan, por lo general, de dos palabras, un verbo seguido de un sustantivo, si bien las aventuras más sofisticadas incluyen oraciones completas. Algunos de los verbos estándar son GET (coger), DROP (soltar), PUSH (empujar), PULL (tirar), THROW (arrojar), LIGHT (encender), KILL (matar), EAT (comer) y DRINK (beber). GO NORTH (ir hacia el norte), por ejemplo, sería la forma normal de especificar un movimiento, si bien la mayoría de los juegos de aventuras permiten abreviar este tipo de órdenes, y así, es muy corriente el empleo de iniciales, como W para GO WEST (ir hacia el oeste).

EXAMINE (examinar) es un verbo esencial; a menudo es el medio para adquirir mayor información. En el ejemplo anterior, EXAMINE SNAKE (examine la serpiente) podría significar: "Se trata de una culebra de hierba", o tal vez: "La serpiente advierte que usted avanza hacia ella y lo ataca". Algunos verbos no requieren un sustantivo.

INVENTORY (inventario) se emplea para decirle lo que está llevando. Algunos objetos se pueden colocar dentro de otros (agua en una botella o un hacha en un saco, por ejemplo), mientras que otros se pueden llevar encima (por ejemplo, un anillo o una capa).

Con frecuencia se emplea SCORE (puntuación) para que el jugador sepa qué progresos ha hecho en alcanzar los objetivos. Digitar HELP (ayuda) puede facilitar una pista para superar una dificultad, pero generalmente le aconseja que siga intentándolo. En ocasiones, en respuesta a una orden determinada, se le puede decir al aventurero: "¡No puede hacer eso... todavía!", lo que implica que una combinación verbo-sustantivo dará un resultado concreto pero no en ese momento y, quizá, no en ese lugar. Una parte del desafío consiste en averiguar los verbos y los sustantivos que son relevantes para la aventura. Las palabras y las combinaciones que ésta no reconoce suelen dar como respuesta algo como "No le comprendo".







Ian McKinnell

En busca del tesoro

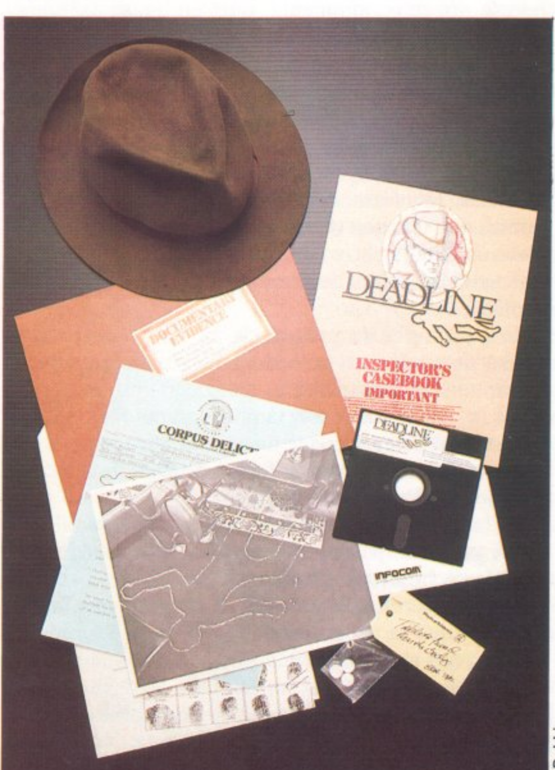
El Hobbit es un juego de aventuras con gráficos para el Sinclair Spectrum y toma su nombre de la novela de J. R. R. Tolkien, un ejemplar de la cual se proporciona con la cassette. Usted asume el papel de Bilbo, viajando a través de Middle Earth y encontrándose con muchos de los personajes y situaciones del libro, en busca del dragón y su tesoro

La mayoría de los editores de aventuras facilitan hojas de pistas para quienes se quedan trabados, ofreciendo sutiles claves para seguir adelante.

Algunos juegos de aventuras son demasiado largos para la memoria del ordenador. Para superar este inconveniente se proporcionan en disco; el programa principal se carga en la memoria al principio y los textos y gráficos se toman del disco o se devuelven a él a medida que se los va necesitando. Pero gracias a las inteligentes técnicas para comprimir grandes cantidades de texto en una memoria restringida, ahora un juego de aventuras largo se puede almacenar en la memoria de un ordenador, lo que significa que estas aventuras se pueden suministrar en cinta de cassette.

Existen aventuras para virtualmente toda clase de micros. Cualquiera le proporcionará una grata experiencia y puede ser el comienzo de un hobby.

¡Felices aventuras!



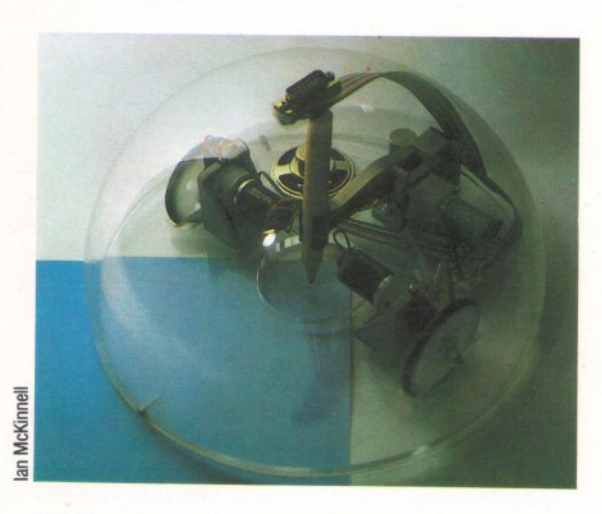
Ian Dobbie

Resolviendo un caso
Deadline (Plazo fatal) es una variante temática de las aventuras. Usted encarna el papel de un detective que debe esclarecer un asesinato. Su archivo contiene, por ejemplo, los resultados de la autopsia, algunas píldoras que se hallaron junto al cuerpo y otras notas relativas al caso. El juego no se desarrolla en tiempo real (dos semanas se considera un muy buen tiempo), pero cada acción que usted empreda, como ir de un cuarto a otro o interrogar a un sospechoso, representa unos minutos vitales que se descuentan del tiempo límite de 12 horas que se le ha dado para resolver el caso

Micromundos

La mayoría de los programas educativos con ordenadores no son sino libros de texto electrónicos. El LOGO es diferente: se vale de un ordenador para crear un "entorno de aprendizaje"

La tortuga que camina
La tortuga se diseñó como un dispositivo con el cual pensar, particularmente durante el ciclo de aprendizaje de geometría y relaciones espaciales. Cuando los niños no están seguros acerca de cómo instruir a la tortuga para que realice una maniobra determinada, tienden a asumir el papel de la tortuga y gatear por el suelo obedeciendo las instrucciones en LOGO. Esto convierte el aprendizaje en una experiencia mucho más "real"


Ian McKinnell

Desde que el micro apareciera por primera vez, en 1977, los educadores con amplitud de miras comprendieron rápidamente el enorme potencial que ofrecía como medio auxiliar de enseñanza en la escuela. En la actualidad la mayoría de las escuelas de los países desarrollados poseen al menos una máquina y en muchas de ellas la informática es una materia más de estudio. A pesar de ello, la incidencia del micro en los métodos de enseñanza tradicionales ha sido ínfima.

La prueba más evidente a este respecto es la gama de programas educativos para ordenadores personales que se ofrece comúnmente, que, en líneas generales, denota una notable falta de imaginación. La mayor parte de estos programas se pueden describir como "libros de texto electrónicos", en los cuales el ordenador le presenta al alumno una serie de "cuadros" en la pantalla (equivalentes a las páginas de un libro de texto) y luego comprueba en qué medida el alumno ha asimilado la información, a través de una serie de preguntas tipo test que deben ser contestadas mediante la elección de una entre varias respuestas, que el ordenador califica automáticamente sin la intervención del profesor.

Los paquetes de programas de este tipo son muy fáciles de escribir en un ordenador personal y ofrecen la ventaja de acompañar el texto con gráficos a todo color (y, en ciertos casos, también animados). Sin embargo, esto no es más que una automatización del método tradicional, y no una forma original de aplicar el potencial del microordenador.

El lenguaje LOGO presenta una perspectiva diferente. A partir de la obra del profesor Seymour Papert, del Massachusetts Institute of Technology, el LOGO se define como "una filosofía de educación y una familia de lenguajes de programación para ordenadores concebida para ayudar a la realización de esa filosofía".

Muchas personas han considerado erróneamente al LOGO como un simple lenguaje de programación y han comparado sus órdenes y configuraciones con las del BASIC, llegando a la conclusión de que el LOGO es un lenguaje mucho mejor para los principiantes. Este razonamiento se aparta de la verdadera cuestión. Papert jamás tuvo la intención de crear un sistema destinado a enseñar al niño a programar. Él lo concibió como un entorno donde el alumno pudiera aprender diversos temas, un ambiente en el que, en realidad, pudiera descubrir una manera de aprender.

Gran parte de esta filosofía deriva del eminente filósofo suizo de la educación Jean Piaget, quien afirmó que el niño, en un entorno adecuado, era capaz de aprender cualquier tema por sí mismo del mismo modo que aprende a andar y a hablar. Sin embargo, la obra de Piaget era completamente teórica, y Papert se abocó a la tarea de crear un ambiente práctico para las teorías de aprendizaje de Piaget.

Que los métodos de educación tradicionales no cumplen este objetivo se hace evidente a partir del hecho de que la mayoría de los adultos teme aprender y no disfruta con la idea de tener que abarcar nuevas áreas de conocimiento. En su libro *Mindstorms — children, computers and powerful ideas (Confusión mental: niños, ordenadores y conceptos eficaces)*. Papert afirma que el síntoma más común de esta actitud del adulto es el miedo generalizado a las matemáticas o, como dice él, la "matematofobia".

Una de las causas de esta postura reside en que la mayor parte de estas disciplinas se imparten de la misma forma, en tanto que sus aplicaciones son totalmente diferentes. Al niño se le enseña, por ejemplo, a multiplicar en la misma forma en que aprende las capitales del mundo: mecánicamente. El proceso de aprendizaje se divorcia de lo que se está aprendiendo, cuando ambas cosas deberían ser inseparables.

Para el propio Papert, el aprendizaje de un conocimiento nuevo, ya sea volar, cocinar o aprender un idioma extranjero, es como practicar una afición. Él atribuye esta actitud suya a su infancia, cuando descubrió, a muy temprana edad, cómo funcionaban las ruedas dentadas y aplicaba este concepto cada vez que se enfrentaba a un nuevo problema. También Albert Einstein solía decir que cuando se encontraba con algo que no comprendía, lo descomponía en conceptos que había aprendido antes de los cinco años.

Estas eficaces ideas se incorporaron al LOGO, como se puede ver en nuestro ejemplo del LOGO en acción. La primera característica importante de este lenguaje es la tortuga, que fue diseñada como "un dispositivo con el que pensar" del mismo modo en que Papert usaba las ruedas dentadas cuando era un niño. Para los niños pequeños, la tortuga asume la forma de un robot móvil diseñado especialmente (véase p. 176), que está conectado a un micro y que se puede desplazar por el suelo digitando órdenes en LOGO. Normalmente la tortuga lleva un lápiz para dibujar formas en el suelo y también puede estar dotada de un pequeño altavoz y de detectores de colisión para guiarla a través de un camino con obstáculos.

Los niños, por lo general, luego de emplear las tortugas móviles, pasan a utilizar las tortugas de pantalla, formas que se pueden mover a través de la pantalla del

Aprendiendo la curva

Este ejemplo ficticio pero típico muestra cómo el LOGO estimula a un grupo de niños a resolver problemas con los cuales nunca antes se habían encontrado.

```
TO CURVE
REPEAT 80
  FORWARD 1
  RIGHT 1
END
CURVE
```

—Si queremos pétalos necesitamos dibujar una curva.
—Pero la tortuga siempre se mueve en línea recta.
—¿Y qué pasaría si la obligáramos a desplazarse una distancia corta, luego la hiciéramos girar sólo un poquito, luego otra vez en línea recta, y así varias veces? Eso sería como una curva.
—De acuerdo; la distancia más pequeña es uno, y el ángulo más pequeño es uno. Hagámosla hacer eso ochenta veces.

```
CURVE
CURVE
```

—¡Eh!, eso es justo lo que queríamos.
—Dos de ellas harán un pétalo; probemos.

```
TO PETAL
CURVE
RIGHT 90
CURVE
END
PETAL
```

—Eso no tiene nada que ver con un pétalo. ¿Qué pasó?
—Que siguió a partir de la última curva: le deberíamos haber dicho que fuera en otra dirección.
—Pero, ¿cuánto la hacemos girar?
—Probemos con noventa; eso suele dar resultado.
—Y hagamos una palabra nueva: PETAL (pétalo), para ahorrar tiempo.

```
TO PETAL
CURVE
RIGHT 100
CURVE
END
PETAL
```

—Eso está mejor, pero noventa grados no era suficiente. ¿Qué hacemos ahora?
—Pensemos algo y veamos si funciona, en vez de hacer suposiciones.
—Sí; recuerda que aprendimos que si la tortuga gira hacia la derecha alrededor de algo gira a través de un total de trescientos sesenta grados.
—Bien, sabemos que gira ochenta en la primera curva; entonces tiene que girar ochenta en el camino de vuelta, o sea ciento sesenta grados.
—Dejando doscientos para girar en la punta del pétalo.
—No, porque para volver a la posición desde donde empezó, necesitará girar también en la otra punta.
—Entonces deberíamos probar con la mitad de doscientos.

```
PETAL
PETAL
PETAL
PETAL
```

—Magnífico: con cuatro como éste haremos una flor.



```
TO FLOWER
REPEAT 4
  PETAL
  RIGHT 10
END
FLOWER
```

—No salió muy bien que digamos; está dispareja.
—Nos olvidamos de poner algunos giros entre los pétalos.
—Pero entonces, ¿por qué no los dibujó uno encima del otro?
—Porque después de dibujar un pétalo, la tortuga queda mirando cien a la izquierda de donde empezó originalmente.
—De modo que cada pétalo gira cien hacia la izquierda cada vez.
—Eso está bastante bien: lo que necesitamos es noventa, así que agreguemos un giro de diez hacia la derecha entre cada pétalo.



```
FLOWER
RIGHT 180
FORWARD 100
RIGHT 180
PETAL
```

—¡Al fin! Gira la tortuga para que podamos dibujar el tallo.
—Cien debería ser suficiente.
—Pongamos una hoja en la punta; puede ser igual a un pétalo.
—Pero esta vez recuerda el ángulo: la tortuga ha de dar vuelta hacia la derecha.



Liz Dixon

ordenador. La tortuga es un eficaz dispositivo con el cual los niños pueden aprender los conceptos básicos de las relaciones espaciales, que les permitirán luego iniciarse en la geometría avanzada.

El control de la tortuga es, sin embargo, sólo una pequeña aplicación del LOGO, pero es el aspecto más promocionado a nivel publicitario porque visualmente es el más interesante. Más importante es el concepto de establecer, a partir de ideas sencillas, ideas más sofisticadas y, a la inversa, descomponer grandes problemas en problemas más pequeños del tipo de los que ya se han asimilado previamente.

Estos procesos se pueden ver con toda claridad a través de la conversación imaginaria entre un grupo de niños que están aprendiendo a instruir a la tortuga para que dibuje una flor (véase recuadro). Comienzan con sólo tres órdenes disponibles: FORWARD (adelante), que hace que la tortuga se mueva hacia adelante hasta una distancia calculada; RIGHT (derecha), que obliga a la tortuga a describir un ángulo determinado, y, por último, REPEAT (repetir), que repite una cierta cantidad de veces las líneas que hay instrumentadas en el programa.

A partir de estas ideas fundamentales los niños construyen primero una "herramienta" (un programa) para dibujar una curva (TO CURVE... END). Toda esta secuencia se puede producir ahora simplemente digitando CURVE. Siguiendo el mismo proceso, después de experimentar y de un aprendizaje posterior, se define una orden PETAL (pétalo), que se vale de la orden CURVE. Finalmente se desarrolla una orden FLOWER (flor), que dibujará la imagen completa.

El LOGO no es el único lenguaje que incorpora este tipo de estructuras (el FORTH es otro de ellos, véase p. 150), pero es el único diseñado para que lo utilicen niños pequeños. Prescinde de muchas de las formalidades y procedimientos asociados con la programación en otros lenguajes. De hecho, se trataba de que el niño no fuera consciente de estar programando un ordenador, sino sólo de estar resolviendo un problema.

En algunas situaciones de aprendizaje, el alumno no se confunde ni siquiera a este nivel. El maestro establece una serie de eficaces herramientas utilizando el LOGO, todas las cuales se refieren a un tema o área de conocimiento determinados. Al niño se le permite luego explorar el tema empleando las herramientas y descubrirlo por sí mismo. Estas áreas se denominan "micromundo": entornos limitados en los cuales se utiliza el ordenador para imitar algo del mundo real o alguna área de conocimiento.

Probablemente el mejor ejemplo de lo que es un micromundo sea el modelo en LOGO de la física de Newton. Aunque la primera ley de Newton dice que sin la influencia de fuerzas externas un cuerpo se seguirá moviendo en línea recta a una velocidad constante, las mentes de los pequeños observan que en el mundo real todo va moviéndose más despacio. Esto determina que el aprendizaje se bloquee. No obstante, utilizando el LOGO se puede construir un micromundo en el que todo se comporte de acuerdo con las leyes de Newton y, con la ayuda de herramientas con las cuales se empujan los objetos a través de la pantalla, los niños aprenden rápidamente y por sí mismos las tres leyes de Newton.

El LOGO es un concepto poderoso que vale la pena aprender mediante un ordenador personal. Las tortugas móviles no son baratas. Están comenzando a salir al mercado versiones de LOGO que emplean tortugas de pantalla para diversos ordenadores personales.

La era de los portátiles

A medida que aumenta la sofisticación de los ordenadores, se los puede comprimir más, convirtiendo en realidad el micro portátil



Ian Dobbie

No es un juguete para ejecutivos
Los ordenadores portátiles como el Epson HX-20, con hasta 32 Kbytes de memoria, periféricos incorporados y una amplia gama de software, han aumentado de manera considerable la cantidad de información de que puede disponer un ejecutivo de empresa, independientemente de en qué lugar se halle. Cada microcassette puede retener hasta 230 Kbytes (quizá 40 000 palabras, equivalente a una amplia base de datos), mientras que los programas estándar de tratamiento de textos y hoja electrónica le permiten llevar a cabo su trabajo casi en cualquier parte: ¡hasta en un taxi!

El desarrollo del micro portátil, o nanocomputador, tal como lo conocemos hoy en día, provino de una doble vertiente. Una de éstas fue el aumento de tamaño de las calculadoras de bolsillo, como en el caso de la Sharp PC1251 y la Casio FX700P. La otra fue un proceso evolutivo de miniaturización, a partir del cual se produjeron máquinas como los ordenadores personales Tandy TRS80 Modelo 100 y el Epson HX-20 (véase página 169).

Estos adelantos fueron consecuencia directa del desarrollo de chips comprimidos en mayor medida, que permitieron incluir muchísima más información en el mismo espacio físico.

Con el advenimiento del microprocesador de un solo chip, en 1972, se hizo teóricamente posible construir un ordenador completo en una caja de tamaño no mayor que el de un paquete de cigarrillos. Sin embargo, las dimensiones de la pantalla de visualización y el principal medio de acceso (el teclado) impusieron limitaciones prácticas a esta miniaturización.

No obstante, las calculadoras de bolsillo se fueron haciendo más pequeñas y en la actualidad existen relojes digitales que funcionan también como calculadoras, para lo cual requieren un puntero o una aguja incorporada con esa finalidad para operar las "teclas". Pero en ellos resulta bastante difícil realizar una sencilla operación aritmética, e incluso el más entusiasta renunciaría a la tarea de dar entrada a un programa de 50 líneas en BASIC. De manera que es poco probable que los ordenadores personales lleguen a ser tan pequeños como un reloj de pulsera. Pero han llegado a ser del tamaño de una calculadora.

Una vez que las calculadoras de bolsillo se hicieron programables, si bien en su propia notación de programación, faltaba sólo un paso para la incorporación de un lenguaje de alto nivel, y el más adecuado era el BASIC. Casi simultáneamente, los fabricantes comenzaban a utilizar una RAM "estable" (una clase de memoria que, reteniendo una pequeña carga eléctrica, no pierde su contenido al interrumpirse la alimentación eléctrica) y un generador de caracteres más amplio para permitir la visualización de caracteres alfabéticos además de los numéricos.

Dispositivos de precio razonable, como los ordenadores de bolsillo Sharp y Casio, que poseen una gama de órdenes en BASIC comparable a la de los ordenadores personales y capacidad de memoria similar, están invadiendo rápidamente el mercado de las calculadoras programables. Diseñados para que se puedan llevar cómodamente en el bolsillo (incluso en el de una camisa), algunos ofrecen la posibilidad de poderlos conectar de manera directa, mediante interfaces, tanto con una impresora como con una grabadora de cassette. Sin duda, en el caso de que exista suficiente demanda, el próximo paso será incorporar un dispositivo para comunicaciones que permita transferir información a través de las líneas telefónicas. La gama Casio, sobre todo, resulta muy conveniente para científicos e ingenieros, ya que retiene la amplia gama de funciones matemáticas y científicas que hicieron tan atractivas las calculadoras de esta marca.

Existen modelos con impresora incorporada que incluyen interface para cassette, como el PC1251 de Sharp.

El siguiente paso trascendental lo representan nanocomputadores como el Epson HX-20, el Tandy 100 y el NEC 8220. Estos micros ofrecen una completa versión de BASIC (Microsoft en todos los casos), una RAM utilizable de entre 16 y 64 Kbytes, una LCD (*Liquid Crystal Display*: visualización en cristal líquido) incorporada de dimensiones razonables (20 × 4 caracteres en el Epson, 40 × 8 en los otros dos) y la posibilidad de conectarlos a una amplia gama de periféricos estandarizados a nivel industrial.

Por cierto, si se los acopla a un monitor regular (para ello quizá sea necesario adquirir una interface especial), se tendrá un micro comparable, en cuanto a capacidad y rendimiento, a un ordenador personal convencional y al mismo precio. La diferencia fundamental radica en el hecho de que estas máquinas dependen de su propia fuente de alimentación eléctrica interna, lo cual les confiere una absoluta portabilidad.

El Epson, por ejemplo, posee pilas de cadmio-níquel incorporadas de las cuales una carga completa dura hasta 50 horas, y el sistema está diseñado de tal manera que cuando las pilas se están agotando él

mismo se interrumpe para preservar la información esencial.

En comparación, el NEC y el Tandy utilizan un CMOS (*Complementary Metal Oxide Semiconductor*: semiconductor de óxido metálico complementario), que requiere mucha menos energía y les permite funcionar con pilas normales tan ligeras como un lápiz.

Al Epson se le consideró en un principio como una "ayuda para ejecutivos", pero es más adecuado para utilizarlo fuera de la oficina que dentro de ella. Es perfectamente apropiado para captación de datos (recolección de información) en la fábrica, en lugares alejados o incluso mientras se camina por la calle, para ser procesados luego en la oficina.

En la práctica el Epson ofrece diversas maneras de transmitir datos. Usado con accesorios patentados, se puede emplear como un terminal de télex; a través de un modem es posible conectarlo a otra máquina vía línea telefónica; o, lo que resulta más sencillo y más barato, los datos se pueden almacenar en cinta de cassette y enviarlos luego por correo.

Es precisamente esta adaptabilidad y versatilidad lo que hace que estas máquinas les resulten tan atractivas a personas que nunca antes habían considerado la posibilidad de emplear ordenadores en su trabajo cotidiano. Un vendedor, por ejemplo, que realice veinte o más visitas al día simplemente para recoger los pedidos habituales, en vez de rellenar formularios impresos puede utilizar una máquina portátil de esta capacidad y potencial. El programa para procesar los pedidos le proporcionaría una lista de productos y cuando el cliente le hiciera su encargo él sólo apuntaría las cantidades. Al final de cada visita la información del pedido se volcaría en la cinta y, concluida la jornada, el vendedor enviaría la cinta por correo hacia la oficina central o bien transmitiría la información directamente a través del sistema informático principal de la empresa.

La propia impresora del micro le proporcionaría al vendedor una copia de sus pedidos del día y al cliente una confirmación inmediata.

Versiones ligeramente más sofisticadas pueden incluir, asimismo, los niveles de stock y advertir cuando éstos disminuyen peligrosamente, si bien en el caso de que varios vendedores trabajaran al mismo tiempo, se habría de conectar el micro con el ordenador principal durante cada visita. Esto no representaría ningún problema si se dispusiera de un acoplador acústico barato y liviano. En este punto nuestro portátil barato se convierte en un terminal interactivo, que permite que se pueda interrogar a toda la base de datos de la empresa obteniendo una respuesta inmediata.

Aparte de las evidentes ventajas que ofrece el acceso inmediato a la información actualizada, el ahorro de tiempo y costo que representa esta clase de entrada de datos en líneas puede amortizar en cuestión de meses el desembolso que supone un sistema portátil completo.

Así como los ordenadores de bolsillo más pequeños están desplazando a las calculadoras personales, los nanocomputadores se están desmarcando de los dispositivos de captura de datos. Estos "terminales mudos" a pilas (que no se pueden programar) han estado a la venta durante cierto tiempo en algunos países, pero no han llegado a popularizarse debido a su elevado precio y a las dificultades que entraña su funcionamiento.

Dados su adecuado software, su capacidad para conectarse a impresoras y monitores de oficina y, quizá, a unidades de disco o cintas de cassette con un acceso más rápido, probablemente los ordenadores portátiles sí se popularizarán. Las máquinas de este tipo proporcionan el primer mercado real para la memoria de burbuja: una memoria estable y autocontenida basada en un chip radicalmente distinto, que en un pequeño espacio ofrece una enorme capacidad de almacenamiento. Aplicaciones clásicas, como el Sharp PC5000, proporcionan 128 Kbytes de almacenamiento por cada cartucho enchufable, con un tiempo de acceso más rápido que el del disco. Resulta muy atractiva la posibilidad de unas máquinas pequeñas que posean un millón de bytes de memoria incorporados.

Esta capacidad de almacenamiento se encuentra disponible en disco en un tercer tipo de máquinas portátiles, representadas por el Osborne Executive, el Ajile Hyperion y el Portico Miracle. Mucho más caros que el Epson HX-20 o el TRS100, estos ordenadores pueden describirse con mucha propiedad como transportables, porque requieren enchufarse a un tomacorrientes. Existen paquetes de pilas para algunas de estas máquinas, pero es muy raro que superen las dos horas de vida útil.

La especificación de estos ordenadores suele incluir unidades gemelas de discos flexibles, monitor de televisión, teclado desmontable y un sistema operativo como el CP/M (*Control Program for Microprocessors*: programa de control para microprocesadores).

Máquinas para todos los gustos
(En el sentido de las agujas del reloj, desde arriba, derecha.) El Epson HX-20, primero de la nueva generación de ordenadores portátiles, ofrece hasta 32 Kbytes de RAM y una amplia gama de periféricos. Con menos potencia (y menor precio) figuran el Sharp PC1251, que aparece con impresora y unidad de microcassette, y el Casio FX700P, que utiliza una grabadora de cassette estándar. Después de haber alcanzado un buen lugar en el mercado de ordenadores "transportables", el Executive, segundo modelo de la Osborne Computer Company, ofrece una pantalla mayor y otros refinamientos



Ian McKinnell

Un procesador de textos
El concepto de un "cuaderno de notas electrónico" ha dado un paso más hacia adelante con la introducción del *microwriter*, un procesador de textos que se puede manejar con una sola mano. Las seis teclas se pulsan en diferentes combinaciones para obtener el alfabeto completo. La salida puede ser directa a una impresora, con algunas órdenes de formato para que el usuario componga el texto, o bien a un procesador de textos de escritorio para su almacenamiento



Ian McKinnell



Herramientas para el comercio
Los ordenadores portátiles baratos han encontrado su sitio en áreas anteriormente inaccesibles para los sistemas de procesamiento de la información

Los periodistas pueden dar entrada en su máquina portátil a las preguntas que desean formular durante las entrevistas, introducir junto a ellas las respuestas y luego editar el artículo, tal vez en el tren, rumbo a casa

Los farmacéuticos disponen de un modo sencillo y barato de etiquetar sus preparados, que también facilita la tarea de controlar las existencias

Los vendedores pueden hacer presupuestos a simple vista, incluso para cálculos complicados, como la instalación de un sistema de calefacción central o una póliza de seguro de vida

El control de existencias es muy importante aun en las tiendas pequeñas, pero en la industria, donde los valores de stock implican muchos millones de pesetas, resulta imprescindible mantener un registro exacto. Los micros portátiles, con 32 o 64 Kbytes de RAM, almacenamiento en cinta de cassette y configuración para comunicaciones, resultan ideales para este fin

David Higham

El primero que salió al mercado británico fue el modelo 1 de Adam Osborne, que ofrecía una gama de software completa, incluyendo BASIC, Wordstar y Supercalc y programas de CP/M por menos de 2 000 libras (unas 450 000 pesetas).

Su punto débil era la pantalla, demasiado reducida (4 × 3,5 pulgadas, o sea, 100 × 85 mm), por lo cual los caracteres eran más bien pequeños y hacía necesaria la utilización del movimiento de pantalla horizontal para conseguir un tamaño legible de visualización cuando se empleaba un paquete de tratamiento de textos como el Wordstar. Sin embargo, los usuarios se acostumbraron a ella rápidamente y el Osborne 1 se convirtió en líder del mercado británico.

La competencia se apresuró a ofrecer más configuraciones, especialmente en términos de potencia y velocidad del microprocesador, y de la capacidad de almacenamiento en disco, hasta que esta clase de máquinas llegó a resultar imposible de distinguir, excepto en su aspecto, de los sofisticados sistemas para oficinas y empresas. En muchos casos ofrecen compatibilidad absoluta con máquinas específicas, en particular el ordenador personal IBM, lo cual quizá favorezca la utilización de ellas como un "segundo ordenador".

La verdadera diferencia entre estos portátiles y los sistemas para oficina radica en su estructura. Un programa de aplicaciones para ejecutar en un sistema normal para oficina, podrá realizarse en un portátil que posea un microprocesador similar. La única limitación continúa siendo, probablemente, el tamaño de la pantalla del monitor y, en consecuencia, de los caracteres individuales.

Todo esto ha sido posible gracias al desarrollo del microprocesador de un solo chip. Las limitaciones en cuanto al nivel de acceso de entrada y salida son consecuencia de nuestra dependencia del teclado como principal dispositivo de entrada y del tubo de rayos catódicos como principal dispositivo de salida. Pero, ¿y si sugiriésemos el reconocimiento y la síntesis de voz como una alternativa? No existiría entonces razón alguna por la cual un micro de potencia similar a la del Ajile Hyperion, por ejemplo, pudiera ocupar más espacio que un Sony Walkman; y, con toda probabilidad, si dispusiéramos de la tecnología adecuada, tendría exactamente ese aspecto, con la incorporación de un micrófono en miniatura.

En el momento actual, cuando los avances tecnológicos se están sucediendo a tanta velocidad, es probable que se produzcan desarrollos en este campo, incluso aunque no incurramos en la fantasía de imaginar la entrada y salida de voz humana. Imaginemos, por ejemplo, una combinación entre el Sinclair ZX Spectrum, el televisor de pantalla plana de Sinclair y un microdisco, proporcionándole energía al sistema a través de una pila recargable de cadmio-níquel: ¿por qué no podemos juntar todo esto? Aunque quizá sea la Sinclair la que lo haga por nosotros...

Debido a su pequeño tamaño, el ZX Spectrum es una máquina muy adecuada para esta aplicación, pero la mayoría de los ordenadores operan en base a la clase de voltaje obtenible de pilas secas normales: existen muy pocas razones por las cuales no se pueda pensar que casi cualquier ordenador podría convertirse en portátil de esta manera.

El ACT Apricot tipifica una tendencia interesante; se trata de un sistema normal para oficina, si bien de pequeñas dimensiones. Dejando de lado el monitor, el Apricot se convierte en un sistema portátil. Por supuesto, "portabilidad" es un término muy vago. Muchos micros se describen como portátiles cuando simplemente están equipados con un asa para llevarlos, y jamás se construyeron con esa finalidad.

Con la introducción de una amplia gama de micros portátiles, la industria ha dado un gran paso hacia la consecución de su verdadero potencial: hacer accesible a todos el poder de la informática a un bajo costo y de forma que no se requieran conocimientos o experiencia previos. Hemos de admitir que aún ha de pasar algún tiempo antes de que se cumplan por completo estos criterios; pero ahora el objetivo está más cerca.

Epson HX-20

Este ordenador no sólo se puede llevar a cualquier sitio, sino que le permite al usuario hacer su trabajo de oficina mientras viaja

El Epson HX-20 fue el primer ordenador auténticamente portátil. Como es totalmente programable en BASIC, posee más aplicaciones que las calculadoras más sofisticadas. Aun así, su diseño autocontenido y sus menos de dos kilogramos de peso total significan que se lo puede llevar en el interior de un maletín.

Muchas máquinas de mayores dimensiones que hasta entonces se vendían como "portátiles", hubieron rápidamente de empezar a anunciarse como "transportables". Un amplio sector del mercado de usuario de microordenadores ha comprado el HX-20: aficionados a la informática personal, hombres de negocios e ingenieros.

Debido a que el HX-20 no se basa en un diseño estándar sino que el suyo es nuevo por completo, el software disponible es todavía relativamente escaso. No obstante, es una máquina ideal para aprender a programar en BASIC. Con frecuencia el comprador tiene la intención de escribir un programa para una aplicación específica (y tal vez inusual): un programa de presupuestos para agentes de seguros, un medio auxiliar de navegación para deportistas náuticos y un apuntador de notas para periodistas; todos ellos son programas escritos para el Epson.

La unidad viene con una pantalla de visualización en cristal líquido que puede manejar hasta 4 líneas de 20 caracteres, o gráficos sencillos de una resolución de hasta 120 × 32 puntos. En la mayoría de las aplicaciones esta visualización actúa como una "ventana" que se puede mover mediante las teclas del cursor (señalizadas con flechas) para mostrar cualquier sección de una superficie de texto mucho mayor que maneja el ordenador.

El mecanismo de la impresora incorporada utiliza papel común en un rollo de 5 cm de ancho, sobre el cual puede imprimir hasta 24 columnas de texto.

La unidad de microcassette es un extra opcional, pero normalmente se la muestra como si fuera incorporada debido a que la mayoría de la gente la necesita. El espacio que ocupa se puede utilizar para un software de cartucho transistorizado, aunque por el momento no se ha producido ninguno. La microcassette es superior a la grabadora de cassette corriente. El ordenador detecta la posición de la cinta y se mueve *fast forward* (rápido adelante) automáticamente para hallar el programa o la información requeridos.

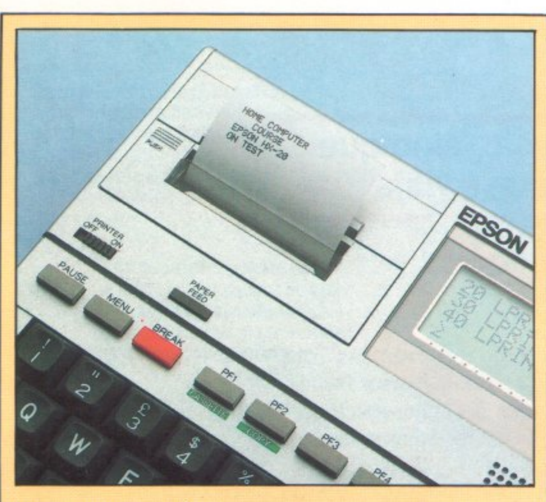
La gama de interfaces situada en la parte posterior y en los laterales de la carcasa refleja la diversidad de las aplicaciones de la máquina; incluso se le puede acoplar un lector de código de barras (véase p. 40). Los 16 Kbytes de RAM se pueden ampliar a 32 Kbytes mediante un accesorio que se coloca en un lateral.

Hay a la venta hardware y software nada caros para que el HX-20 se pueda comunicar a través del teléfono con otra máquina similar o con un ordenador principal, y tener acceso a información central.



Chris Stevens

El teclado del Epson HX-20
Es su completo teclado, de dimensiones similares al de una máquina de escribir, lo que en realidad determina el tamaño del HX-20. Si bien el "tacto" de las teclas es algo distinto del de una máquina de escribir convencional, satisface plenamente a los mecanógrafos al tacto.
Además de los numerales situados en la hilera de teclas superior, al pulsar NUM las teclas U, I, O, J, K, L y M se convierten en un relleno de teclado numérico. Esto permite que se pueda efectuar con mayor rapidez la entrada de grandes cantidades de información numérica.
El movimiento del cursor y otras teclas para edición se halla arriba a la derecha, junto con una tecla SCRN (*scrolling*) para hacer girar la pequeña pantalla hacia arriba y hacia abajo.
Las cinco teclas de función programable (de PF1 a PF5) son físicamente diferentes de las otras teclas y la función de dos de ellas es doble, para controlar la microcassette y para copiar en la impresora el contenido de la pantalla



Impresora incorporada
Las pequeñas impresoras matriciales o de rollo de papel han sido parte integrante de las cajas registradoras (e incluso de las calculadoras) durante bastante tiempo. La casa Epson, que dio sus primeros pasos como fábrica de impresoras, incorporó muy rápidamente esa idea a su portátil HX-20. Esta impresora matricial ofrece una salida tanto de gráficos como de caracteres

Enchufe red
Con el transformador conectado, el HX-20 se alimenta desde la red eléctrica y recarga su batería interna

Botón de borrado

Conexión códigos de barras
Aquí se puede conectar un lápiz óptico para leer códigos de barras, que existen actualmente en muchos comercios

Conexión cassette
Trabaja con una grabadora de cassette corriente e incluye el control del motor. No obstante, su rendimiento no es tan eficaz como el de la microcassette



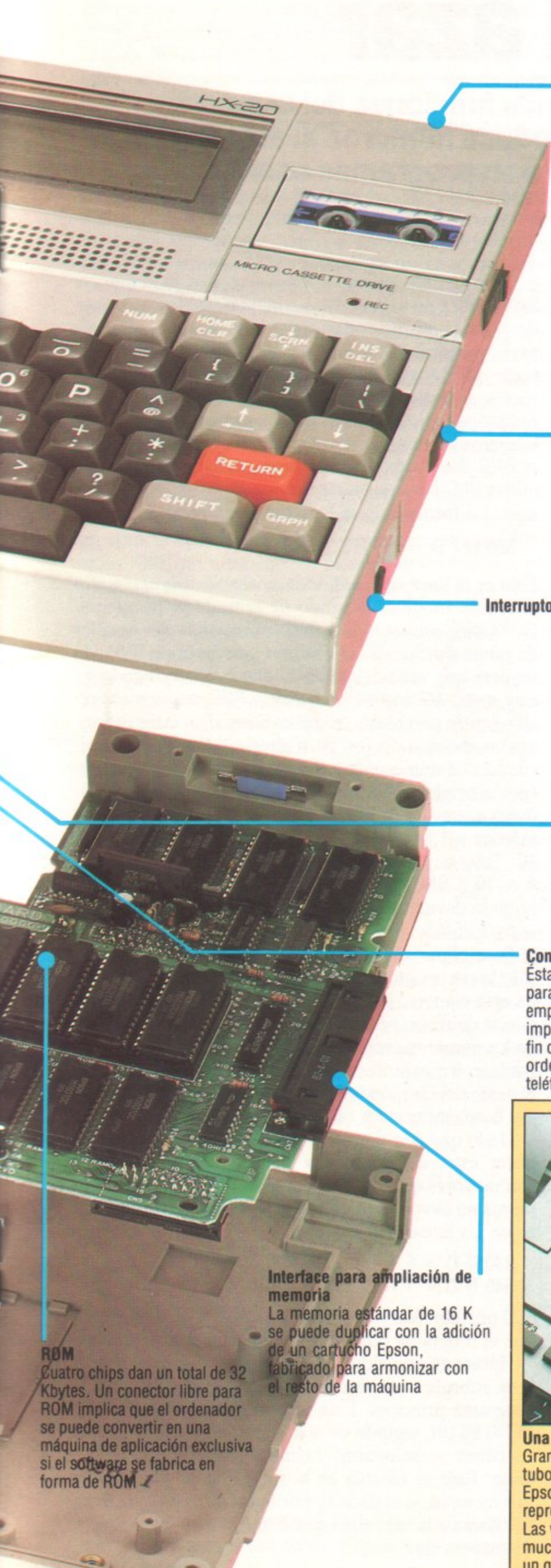
Almacenamiento de información
La grabadora de microcassette incorporada del HX-20, construida alrededor de la clase de cintas de cassette que utilizan las máquinas para dictado de bolsillo, representa un avance significativo respecto a las grabadoras de cassette corrientes, porque el ordenador controla la función *fast wind* (bobinado rápido). Por lo tanto se reduce de manera drástica el tiempo necesario para llegar a un determinado punto de la cinta. Otra ventaja es que la unidad se enchufa directamente en un espacio reservado de la carcasa, sin que medie ningún tipo de cable

Placa de circuito impreso
En nuestra ilustración, el PCB (*Printed Circuit Board*: placa de circuito impreso) se ha invertido por razones de mayor claridad. Normalmente los circuitos miran hacia la cara inferior del ordenador y las puertas separables en el fondo de la carcasa significan que se pueden cambiar los chips más importantes sin tener que desmontar totalmente la unidad

Microprocesadores
Dos microprocesadores 6301 (fabricados por Epson) controlan el ordenador y sus interfaces. Extraordinariamente, cada uno posee 4 Kbytes de ROM y 128 bytes de RAM incorporados, además de los chips exteriores

RAM
El HX-20 viene con 16 Kbytes de RAM estándar, configurados en forma de ocho chips de 2 Kbytes





Disparador de cartucho
Este mando permite retirar entera la microcassette. En teoría este módulo se puede sustituir por un cartucho transistorizado, en el caso de que el usuario desee utilizar la máquina para una sola finalidad

Ajustador de ángulo de visión
Normalmente, para ser legibles, las pantallas de visualización en cristal líquido han de verse desde el ángulo corrector. Haciendo rotar este mando el usuario puede mirar desde cualquier ángulo

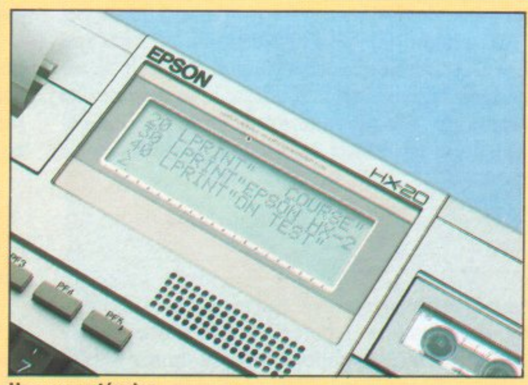
Interruptor de potencia

Interface en serie
A pesar de que la conexión RS232 también es una interface en serie, existe esta otra conexión para activar diversos periféricos que la Epson tiene previsto introducir para el HX-20

Conexión RS232
Ésta es la conexión estándar para interfaces en serie y se emplea para acoplar una impresora o un modem, con el fin de comunicarse con otros ordenadores a través del teléfono

Interface para ampliación de memoria
La memoria estándar de 16 K se puede duplicar con la adición de un cartucho Epson, fabricado para armonizar con el resto de la máquina

ROM
Cuatro chips dan un total de 32 Kbytes. Un conector libre para ROM implica que el ordenador se puede convertir en una máquina de aplicación exclusiva si el software se fabrica en forma de ROM



Una nueva técnica
Gran parte del volumen de un microordenador lo constituye el tubo de rayos catódicos que conforma el corazón del monitor. El Epson HX-20 superó este problema sustituyendo la unidad de representación visual común por una pantalla en cristal líquido. Las visualizaciones de este tipo se han venido usando desde hace mucho tiempo en las calculadoras, pero esta aplicación significa un gran paso hacia adelante en términos de calidad, ya que la matriz de 20 × 4 posee un ajustador de ángulo de visión

EPSON HX-20

DIMENSIONES
290 × 210 × 45 mm

PESO
1,7 kg

CPU
6301 Hitachi

VELOCIDAD DEL RELOJ
2,45 MHz

MEMORIA
16 K de RAM, ampliables a 32 K mediante un accesorio..
32 K de ROM, ampliables a 64 K por el mismo método

VISUALIZACION EN VIDEO
Pantalla en cristal líquido (con ajustador de ángulo de visión) con 4 líneas de 20 caracteres que actúan a modo de ventana de una superficie de hasta 255 × 255 caracteres. Cada pixel de la visualización se puede direccionar, obteniendo una resolución para gráficos de hasta 120 × 32 puntos

INTERFACES
Conexiones para ampliación de memoria, cassette, RS232 (para impresora exterior), serial (para futuro disco), lápiz óptico para códigos de barras

LENGUAJE SUMINISTRADO
BASIC

OTROS LENGUAJES DISPONIBLES
Intérprete de FORTH

VIENE CON
Maletín de plástico moldeado para transportarlo, fuente de alimentación eléctrica, manual de instrucciones

TECLADO
60 teclas tipo máquina de escribir, cinco teclas de función programable y cuatro botones para función exclusiva

DOCUMENTACION
El ordenador viene con dos manuales: un manual de funcionamiento y un manual de referencia de BASIC. Ambos son exhaustivos, pero no poseen índice. La firma Epson optó por incorporar BASIC Microsoft, estándar en la industria, y el manual para el lenguaje probablemente sea adecuado como instrucción general, y guía de referencia.
El manual de funcionamiento incluye información para especialistas y una completa lista de las conexiones de I/O para el RS232 y las interfaces en serie; los niveles de señalización son aceptables, así como las extensas tablas de los mapas de memoria

Números al azar

Siguiendo con nuestro estudio de las funciones del BASIC llegamos a la función RND, que produce números al azar (o casi al azar) para su utilización en juegos o en programas estadísticos

Ahora que ya hemos visto cómo trabajan varias de las funciones de BASIC, examinaremos una de las que más se utilizan: la función RND. RND se usa para generar números al azar. También se emplea en los juegos cada vez que existe un elemento de casualidad.

Lamentablemente, RND es una de las palabras más "variables" del BASIC. La descripción que nosotros hagamos de ella puede diferir respecto a la función que tenga destinado realizar en algunos micros personales. Aclararemos, entonces, las diferencias existentes entre el BASIC utilizado en nuestro curso de programación BASIC y otros BASIC.

La mayoría de nuestros programas se basan en el BASIC Microsoft (o MBASIC). Microsoft es una empresa norteamericana y su BASIC fue uno de los primeros que estuvieron disponibles a nivel masivo. El BASIC es un lenguaje que no está estandarizado oficialmente, pero el de Microsoft se aproxima, dentro de lo posible, a lo que sería una versión estandarizada. Existen muchas versiones elaboradas a partir de la de Microsoft, y esta empresa tiene encomendado producir otras para varios ordenadores populares.

La diferencia fundamental entre el MBASIC y la mayoría de las versiones más recientes es que ahora los ordenadores personales son capaces de realizar excelentes gráficos, facultad que no poseían cuando se desarrolló el MBASIC. Otras versiones de BASIC incluyen generalmente un número de órdenes y sentencias para gráficos. Para obtener el máximo rendimiento de las prestaciones del ordenador, es necesario aprovechar en profundidad su capacidad para realizar gráficos, y ello requiere un exhaustivo estudio del manual por parte del usuario.

De los diversos BASIC que proporcionan los ordenadores personales más populares, probablemente el BASIC del Sinclair (utilizado en el ZX81 y en el Spectrum) y el BASIC del BBC sean los que más difieren del MBASIC. La versión del Texas Instruments (empleada en el TI99/4A) también incorpora una cantidad de diferencias significativas. En la medida de lo posible, en los recuadros de "Complementos al BASIC" le indicamos cómo debe modificar nuestros programas, y se



debe remitir a estos recuadros en caso de que encuentre alguna dificultad al ejecutar los programas.

Como hemos mencionado antes, la función RND varía de una versión a otra. Compruebe en su manual de BASIC qué cometido cumple en su versión. Nosotros vamos a ilustrar su utilización a partir de un sencillo juego de dados. Al igual que en programas anteriores, hemos realizado la mayor parte del trabajo en forma de subrutinas. Esta técnica ofrece la ventaja de hacer que los programas resulten más fáciles de leer, de escribir y de depurar.

El programa principal comienza con la sentencia RANDOMIZE en la línea 20. La mayoría de las versiones de BASIC, pero no todas, necesitan de esta sentencia para "redistribuir" la función RND. En realidad es bastante difícil lograr que los ordenadores produzcan números auténticamente al azar. Sin esta operación de redistribución, se produciría la misma secuencia de números supuestamente al azar cada vez que se aplicara RND. La línea 50 remite luego a una subrutina que utiliza RND para asignarle a la variable D un número al azar. La forma en que la hemos empleado es:

```
320 LET D = INT (10 * RND)
```

Ésta es la línea que, con toda probabilidad, el usuario habrá de modificar cuando dé entrada al programa. En "Complementos al BASIC" se proporcionan detalles de cómo funcionan las distintas versiones de RND, de manera que veamos qué es lo que está sucediendo en este BASIC Microsoft. La función RND usa una expresión (entre paréntesis, como es normal en estas funciones) como una opción para alterar ligeramente la secuencia de números producida. Sin ninguna expresión (por ejemplo, LET A = RND), el valor de A será un número entre 0 y 1. Nosotros no deseamos un número inferior a 1, por lo cual multiplicamos el número por 10. Esto se puede hacer de la siguiente manera: LET A = 10 * RND. Si, en aras de la demostración, RND hubiera devuelto el valor 0,125455, ahora el valor de A sería 1,25455.

Para eliminar la fracción del número y conservar sólo la parte entera, utilizamos la función INT (entero) de esta manera: LET A = INT (10 * RND). Algunas versiones de BASIC permiten especificar el límite máximo de los números generados al azar en la expresión encerrada entre paréntesis después de RND. Por ejemplo, el BASIC del Dragon imprimirá un número entero entre 1 y 6 en respuesta a PRINT RND(6).

Dado que con nuestro BASIC Microsoft no podemos hacer esto, comprobamos si los números devueltos son mayores que 6 o menores que 1, porque esos números no sirven para un juego de dados. Esto se realiza en las líneas 330 y 340:

```
330 IF D > 6 THEN GOTO 320
340 IF D < 1 THEN GOTO 320
```

Si D quedara fuera de los límites entre 1 y 6, GOTO hará que el programa salte hacia atrás y lo intente otra vez.

Habiendo escogido para D un valor al azar entre 1 y 6, la subrutina de tirar los dados vuelve (RETURN) al programa principal. Éste imprime el mensaje SU NUMERO ES UN, seguido de la imagen de un dado. Observe cómo se selecciona la imagen apropiada de un dado. Esto se efectúa en la subrutina de SELECCION. Por ejemplo, si el dado (y por tanto D) es un 1, la línea 410 llama a la subrutina que comienza en la línea 530, de manera que:

```
410 IF D = 1 THEN GOSUB 530
```

Esta subrutina no es más que una serie de sentencias PRINT diseñada para producir gráficos muy rudimentarios en la pantalla. Bien puede ser que el BASIC del ordenador del usuario posea gráficos de pantalla mucho mejores y, si éste es el caso, lo mejor sería reemplazar nuestras subrutinas por las sentencias para gráficos apropiadas.

Una vez que el programa ha elegido un dado al azar para el usuario, repetirá luego el proceso para escoger y visualizar un dado para el ordenador. La parte del programa que decide quién ha ganado se ha incorporado al programa principal; también se la podría haber escrito en forma de subrutina, pero no habría valido la pena, puesto que sólo tiene cuatro líneas. La línea 200 compara M (mis dados) con C (los del ordenador) para ver si son iguales. Si lo son, a la variable alfanumérica S$ se le asignan las palabras ES UN EMPATE. La línea 210 compara para comprobar si M es mayor que C. Si lo es, le asigna a S$ las palabras HA GANADO USTED. La línea 220 comprueba si M es menor que C. En ese caso, le asigna a S$ las palabras HA GANADO EL ORDENADOR. La línea 240 simplemente imprime el resultado y el juego acaba. Aunque este programa es bastante largo, en esencia es muy sencillo. Utiliza solamente una función, RND, no tiene bucles, ni variables subíndice, ni ninguna otra complicación excepto algunas sentencias IF...THEN.

Dado que la función RND es tan variable y que algunas versiones de BASIC (la Microsoft, por ejemplo) requieren la sentencia RANDOMIZE para generar una nueva secuencia de números al azar, ¿existe algún procedimiento para que podamos generar números auténticamente al azar (es decir, impredecibles) sin usar estas funciones? Sí, existen varias técnicas.

Una de las funciones que hasta ahora no habíamos examinado es la función INKEY$ (que corresponde a *inkey-string*). Cada vez que se encuentra con la palabra INKEY$, el programa inspecciona el teclado para ver si se ha pulsado alguna tecla. El programa no espera a que se dé entrada a un carácter, como lo hace cuando se emplea la orden INPUT. Por tanto la orden INKEY$ normalmente se utiliza dentro de un bucle. El programa entonces explora continuamente el teclado, esperando alguna entrada. Por lo general en el bucle se incluye alguna comparación para que éste termine cuando se dé entrada a un carácter apropiado. Esto permite escribir un programa que forme un bucle para contar que concluirá cuando se haya digitado un carácter específico. ¿Qué sucedería si empleáramos este programa?

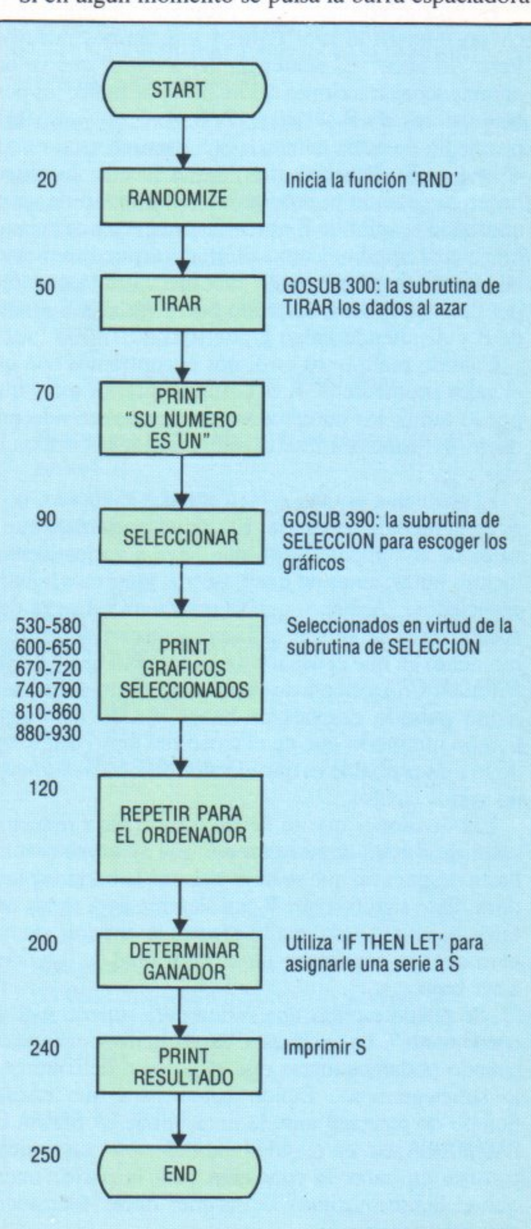
```
10 PRINT "PULSE LA BARRA ESPACIADORA"
20 FOR X = 0 TO 1
30 LET R = R + 1
40 LET A$ = INKEY$
50 IF A$ = "  " THEN GOTO 80
60 LET X = 0
70 NEXT X
80 FOR Q = 0 TO 1
90 IF R < 10 THEN GOTO 130
100 LET Q = 0
110 LET R = R/10
120 NEXT Q
130 PRINT INT (R)
140 END
```

¿Será R un número al azar? Debería serlo, de modo que procedamos a analizar el programa y comprobemos por qué.

La línea 10 imprime la nota PULSE LA BARRA ESPACIADORA. Antes de que podamos responder a esta nota, el programa ha dado entrada el bucle FOR X = 0 TO 1 en la línea 20. Puede que 0 y 1 parezcan límites extraños para el bucle, pero enseguida veremos cómo se utiliza esta estructura. La línea 30 le asigna a la variable R el valor 1 la primera vez que se efectúa el bucle. La línea 40 le asigna a la variable alfanumérica A$ el carácter digitado en el teclado, cualquiera que haya sido. Esto se hace empleando la función INKEY$. Si el usuario pulsara la letra R, a A$ se le asignaría R. La línea 50 compara A$ para ver si es un espacio (que en BASIC se representa como un espacio encerrado entre comillas dobles: " "). Si A$ es un espacio, el programa se bifurca valiéndose de la sentencia GOTO; pero si A$ no es un espacio, el programa continúa por la siguiente línea.

Se trata de la línea 60, que dice LET X = 0. Ahora X es el índice del bucle. La sentencia NEXT X de la línea 70 hace que el programa retorne al comienzo del bucle, en la línea 20. Dado que X se ha reestablecido en 0, el bucle la repite. De esta manera, el bucle FOR X = 0 TO 1 se repetirá indefinidamente, en la medida en que fracase la comparación IF A$ = " "

Si en algún momento se pulsa la barra espaciadora,



Flujo del programa
El diagrama de flujo muestra de forma simplificada las acciones principales que realiza el programa. A la izquierda se indican los números de las líneas correspondientes y, a la derecha, se dan unas breves notas explicativas. No se trata de un diagrama de flujo completo, ya que no incluye muchas de las "decisiones" y bifurcaciones del programa

a A$ se le asignará el carácter que represente un espacio, y el programa se bifurcará hasta la línea 80 y el bucle no se repetirá más.

Pero ¿qué sucede mientras el bucle se repite? A cada repetición del bucle, la línea 30 aumenta el valor de R. La primera vez que se realiza, R se establece en 1, la segunda vez se establece en 1 + 1 y así sucesivamente. Cuando el bucle se interrumpe a raíz de la comparación de A$, podemos leer R para ver hasta dónde hemos contado.

Sin embargo, los ordenadores funcionan a gran velocidad, de modo que, para cuando pulsemos la barra espaciadora, R podría haber llegado a varios centenares. ¿Qué haremos si deseamos que los valores de R se hallen sólo entre 1 y 10? La línea 80 introduce otro bucle que nos permite comparar R y dividirla por 10 en el caso de que sea mayor que 10. En la medida en que R sea mayor que 10, fallará la comparación de la línea 90, el valor de Q se reestablecerá en 0 y se repetirá el bucle. La línea 110 divide por 10 el valor de R, pero el resultado no se imprimirá en tanto no se dé la condición de que el valor de R se haya reducido a un número menor que 10. La línea 30 asegura que el valor de R nunca podrá ser 0.

En teoría, entonces, este programa debería producir un número al azar entre 1 y 9, ambos inclusive. Pero ¿lo hace? La sentencia INT asegura que se han eliminado las fracciones decimales; por tanto, los posibles valores de R serían 1, 2, 3, 4, 5, 6, 7, 8, 9. El promedio de estos números es 5 (porque suman 45, y 45 ÷ 9 = 5). Pruebe y vea lo que sucede. Lo puede hacer ejecutando el programa una cantidad de veces, anotando el valor de R obtenido cada vez y calculando luego el promedio. Como alternativa, puede agregarle algunas líneas al programa para ejecutarlo, pongamos por caso, 100 veces, sumando a otra variable S el valor de R y dividiendo luego S por 100.

Cuando realizamos esto, nos encontramos con que el valor promedio de R es bastante inferior a 5 y que, por lo tanto, los números no pueden haber sido producto del azar. Es interesante analizar las causas de ello.

El problema es que, si bien el BASIC es rápido, no lo es en la medida necesaria. El primer bucle deja que el valor de R aumente hasta que llega a varios cientos, incluso miles, antes de que nosotros pulsemos la barra espaciadora. A menos que se realice un esfuerzo deliberado para modificar el lapso transcurrido entre el momento en que el usuario ve la nota PULSE LA BARRA ESPACIADORA y en efecto la pulsa, lo cierto es que se la habrá pulsado después de haber pasado demasiado tiempo, de modo que en el curso del lapso transcurrido lo más probable es que el valor de R haya aumentado varios cientos.

Las divisiones que se llevan a cabo para reducir el valor de R a un número menor que 10 no se realizan hasta después de que se haya pulsado la barra espaciadora. Esto significa que R casi siempre está en las centenas bajas antes de que se efectúe la división, de manera que, en consecuencia, el valor final de R tenderá a ser bajo.

¿Se puede escribir una rutina para superar este inconveniente? La respuesta es afirmativa, siempre y cuando podamos lograr que el proceso de contar sea lo suficientemente rápido como para que nuestro tiempo de reacción ante la nota PULSE LA BARRA ESPACIADORA sea, en realidad impredecible. La solución consiste en hacer la condición para la parte "mayor que el límite máximo" del primer bucle. Consideremos este programa:

```
10 REM JUEGO DE DADOS -- PROGRAMA PRINCIPAL
20 RANDOMIZE
30 REM SU TURNO
40 REM GOSUB RUTINA DE'TIRAR'
50 GOSUB 300
60 LET M = D
70 PRINT "SU NUMERO ES UN"
80 REM GOSUB RUTINA DE'SELECCION'
90 GOSUB 390
100 PRINT
110 REM TURNO DEL ORDENADOR
120 REM GOSUB RUTINA DE'TIRAR'
130 GOSUB 300
140 LET C = D
150 PRINT "EL NUMERO DEL ORDENADOR ES UN"
160 REM GOSUB RUTINA DE'SELECCION'
170 GOSUB 390
180 PRINT
190 REM  ¿QUIEN GANO?
200 IF M = C THEN LET S$ = "EMPATE"
210 IF M > C THEN LET S$ = "GANO USTED"
220 IF M < C THEN LET S$ = "GANO EL ORDENADOR"
230 REM PRINT RESULTADO
240 PRINT S$
250 END
260 REM
270 REM
280 REM
290 REM
300 REM SUBRUTINA DE TIRAR LOS DADOS AL AZAR
310 REM
320 LET D = INT(10* RND)
330 IF D > 6 THEN GOTO 320
340 IF D < 1 THEN GOTO 320
350 RETURN
360 REM
370 REM
380 REM
390 REM SUBRUTINA DE SELECCION
400 REM
410 IF D = 1 THEN GOSUB 530
420 IF D = 2 THEN GOSUB 600
430 IF D = 3 THEN GOSUB 670
440 IF D = 4 THEN GOSUB 740
450 IF D = 5 THEN GOSUB 810
460 IF D = 6 THEN GOSUB 880
470 RETURN
480 REM
490 REM
500 REM
510 SUBRUTINAS DE 'GRAFICOS'
520 REM
530 PRINT "           "
540 PRINT " ┌───────┐ "
550 PRINT " │       │ "
560 PRINT " │   *   │ "
570 PRINT " │       │ "
580 PRINT " └───────┘ "
590 RETURN
600 PRINT "           "
610 PRINT " ┌───────┐ "
620 PRINT " │     * │ "
630 PRINT " │       │ "
640 PRINT " │ *     │ "
650 PRINT " └───────┘ "
660 RETURN
670 PRINT "           "
680 PRINT " ┌───────┐ "
690 PRINT " │     * │ "
700 PRINT " │   *   │ "
710 PRINT " │ *     │ "
720 PRINT " └───────┘ "
730 RETURN
740 PRINT "           "
750 PRINT " ┌───────┐ "
760 PRINT " │ *   * │ "
770 PRINT " │       │ "
780 PRINT " │ *   * │ "
790 PRINT " └───────┘ "
800 RETURN
810 PRINT "           "
820 PRINT " ┌───────┐ "
830 PRINT " │ *   * │ "
840 PRINT " │   *   │ "
850 PRINT " │ *   * │ "
860 PRINT " └───────┘ "
870 RETURN
880 PRINT "           "
890 PRINT " ┌───────┐ "
900 PRINT " │ *   * │ "
910 PRINT " │ *   * │ "
920 PRINT " │ *   * │ "
930 PRINT " └───────┘ "
940 RETURN
```

```
10 PRINT "PULSE LA BARRA ESPACIADORA"
20 FOR X = 0 TO 1
30 LET R = R + 1
40 IF R > 9 THEN LET R = 1
50 IF INKEY$ = "   " THEN GOTO 80
60 LET X = 0
70 NEXT X
80 PRINT R
90 END
```

En este programa, R nunca puede ser menor que 1 ni mayor que 9. Para cuando se pulse la barra espaciadora (y lo reconozca la función INKEY$ de la línea 50), R tendrá un valor cualquiera comprendido entre 1 y 9, ambos inclusive.

Este programa fue comprobado mil veces y se halló para R un valor promedio de 5,014. Puesto que el promedio perfecto sería 5 y que el margen de error es sólo un 0,28 % superior, esto sugeriría que de hecho el programa genera un número al azar que se aproxima bastante al promedio teórico. La cuestión estriba, por supuesto, en que incluso cuando un programa parece razonable sobre el papel, luego, en la práctica, puede tener algunos fallos imprevistos. Este motivo justifica el que siempre valga la pena realizar una comprobación a conciencia.

Algunos lectores habrán observado que estos programas para números al azar se pueden acortar mediante la utilización de varias sentencias GOTO en lugar del bucle FOR...NEXT. En próximos capítulos de nuestro curso de programación BASIC explicaremos las razones por las cuales hemos preferido evitar las sentencias GOTO.

Complementos al BASIC



En el micro BBC y el Oric-1, elimine la línea 20 y sustituya la 320 por:

```
320 LET D = INT (10 * RND(1))
```

En el Dragon-32, suprima la línea 20 y sustituya la 320 por:

```
320 LET D = RND(6)
```

y elimine las líneas 330 y 340.

RND

En el Lynx, sustituya la línea 20 por:

```
20 RANDOM
```

En el Vic-20 y el Commodore 64, reemplace la línea 20 por:

```
20 LET X = RND(-TI)
```

y sustituya la línea 320 por:

```
320 LET D = INT(10 * RND(1))
```

En el Spectrum, la palabra RANDOMIZE aparece abreviada en el teclado (RAND), pero en la pantalla se leerá RANDOMIZE



En el Oric-1 y en el Lynx, reemplace INKEY$ por KEY$.
En el Vic-20 y el Commodore 64, sustituya la línea 40 por:

```
40 GET A$
```

Luego reemplace la línea 50 por:

```
50 GET A$: IF A$ = "   " THEN GOTO 80
```

En el micro BBC, sustituya INKEY$ por INKEY$(10). El número entre paréntesis indica el tiempo, expresado en centésimas de segundo, durante el cual el sistema aguardará a que se pulse una tecla; así que para obtener una respuesta rápida deberá usar un número bajo, y viceversa

Ejercicios

■ **Función RND** Modifique el último programa del texto para que dé un número al azar entre 1 y 6 (ambos inclusive).

■ **Bucle y promedio** Agregue algunas líneas al último programa del texto con la finalidad de hacer que se repita cien veces y a continuación dé el promedio de los cien resultados.

■ **Sustitución por una subrutina** Sustituya las líneas 50 y 130 del programa principal (es decir, la subrutina de tirar los dados al azar) por un GOSUB que invoque a su "generador de números al azar" existente en el primer ejercicio.

■ **INKEY$** Utilizando la función INKEY$, ¿cómo escribiría usted un programa que leyera cualquier tecla digitada en el teclado e imprimiera: LA TECLA QUE USTED HA PULSADO ES: * (* representa la tecla pulsada por usted).

■ **Bucle de tiempos** Escriba un bucle de tiempos (un bucle "para contar") y utilice la función INKEY$ para hallar cuál es el valor más alto que alcanza una variable al cabo de 10 segundos (necesitará un reloj). Escriba el programa de modo que la última salida impresa diga: EL VALOR DE R AL CABO DE 10 SEGUNDOS ES: * (* representa el valor de R).

■ **Comparaciones IF-THEN** Escriba un programa para juegos sencillo en el cual el ordenador genere un número al azar entre 1 y 100 (ambos inclusive) y el jugador tenga que adivinar de qué número se trata. El jugador dispone de tres oportunidades. Cada vez, el programa le responde con los mensajes SU ELECCION ES DEMASIADO ELEVADA, SU ELECCION ES DEMASIADO BAJA, o HA ACERTADO, FELICITACIONES, o YA NO TIENE MAS OPORTUNIDADES. ¡HA PERDIDO!

Las respuestas, en el próximo capítulo.

Respuestas a los "Ejercicios" de las páginas 148 y 149

Bucles 1

EL VALOR DE A ES 450

Bucles 2

START
STOP

Bucles 3

AHORA EL VALOR DE A ES 160

Bucles 4

ME GUSTA EL BASIC
ME GUSTA EL BASIC
ME GUSTA EL BASIC
:
:

Hasta que borre (RESET) o interrumpa (BREAK) el programa

Bucles 5

ESTOY ALGO TONTO
(15 veces)

Read-Data 1

ESTAMOS COMPARANDO LA SENTENCIA 'READ'
170

Read-Data 2

X = 1
X = 2
:
:
X = 23

Sobre dos ruedas

La robótica se está desarrollando rápidamente. Los robots móviles, como el Buggy BBC, se pueden programar incluso para que detecten obstáculos

Los robots y las tortugas móviles no tienen sólo una función educativa sino que también son una fuente de diversión. Los principios que se aplican para controlar un dispositivo como el Buggy de la BBC son los mismos que rigen para los robots industriales de grandes dimensiones. Aunque en realidad todavía no sirven para realizar los trabajos domésticos, podrían conventirse en la próxima generación en dispositivos auxiliares para el hogar.

Es necesario que los robots se sitúen con exactitud en relación a su entorno, y por esta causa normalmente se accionan mediante motores paso a paso. A diferencia de los motores convencionales, los motores paso a paso no rotan cuando se les suministra energía. A cada impulso de energía que se les aplica, el eje gira una fracción preestablecida de rotación completa. El número de impulsos que se requieren para efectuar una revolución completa depende de cada motor. También se puede controlar el sentido de la rotación. Es posible lograr que un robot o una tortuga móviles se desplacen a través de distancias muy precisas, en cualquier dirección, permitiendo que el ordenador accione los motores por separado. Un robot móvil puede girar en el sitio en que se halla accionando sólo las dos ruedas en sentido contrario.

Sin embargo, resulta igualmente importante que el robot se pueda remitir al ordenador cuando se encuentra con algo. Las colisiones se suelen detectar montando unos parachoques alrededor del cuerpo del vehículo y conectándolos a microinterruptores. Éstos, a su vez, están acoplados a la conexión de entrada del ordenador y la apertura o el cierre de cada interruptor hará que uno de los bits cambie a "0" o a "1".

Con frecuencia se requieren otros tipos de entradas desde el robot. Podría ser útil, pongamos por caso, la capacidad para seguir líneas blancas trazadas sobre un suelo oscuro. Esto se consigue haciendo que disminuya la intensidad de una fuente luminosa situada en el robot, que posee junto a ella un detector para medir la cantidad de luz reflejada. Esta cantidad variará a tenor de la superficie por la cual se desplace el robot a cada momento; se trata más bien de una cantidad analógica antes que de una cantidad digital. El microordenador BBC está equipado con una entrada analógica que permite utilizar directamente un detector de este tipo. En la mayoría de los otros sistemas la señal se ha de convertir en una señal digital para que se la pueda volver a enviar al ordenador.

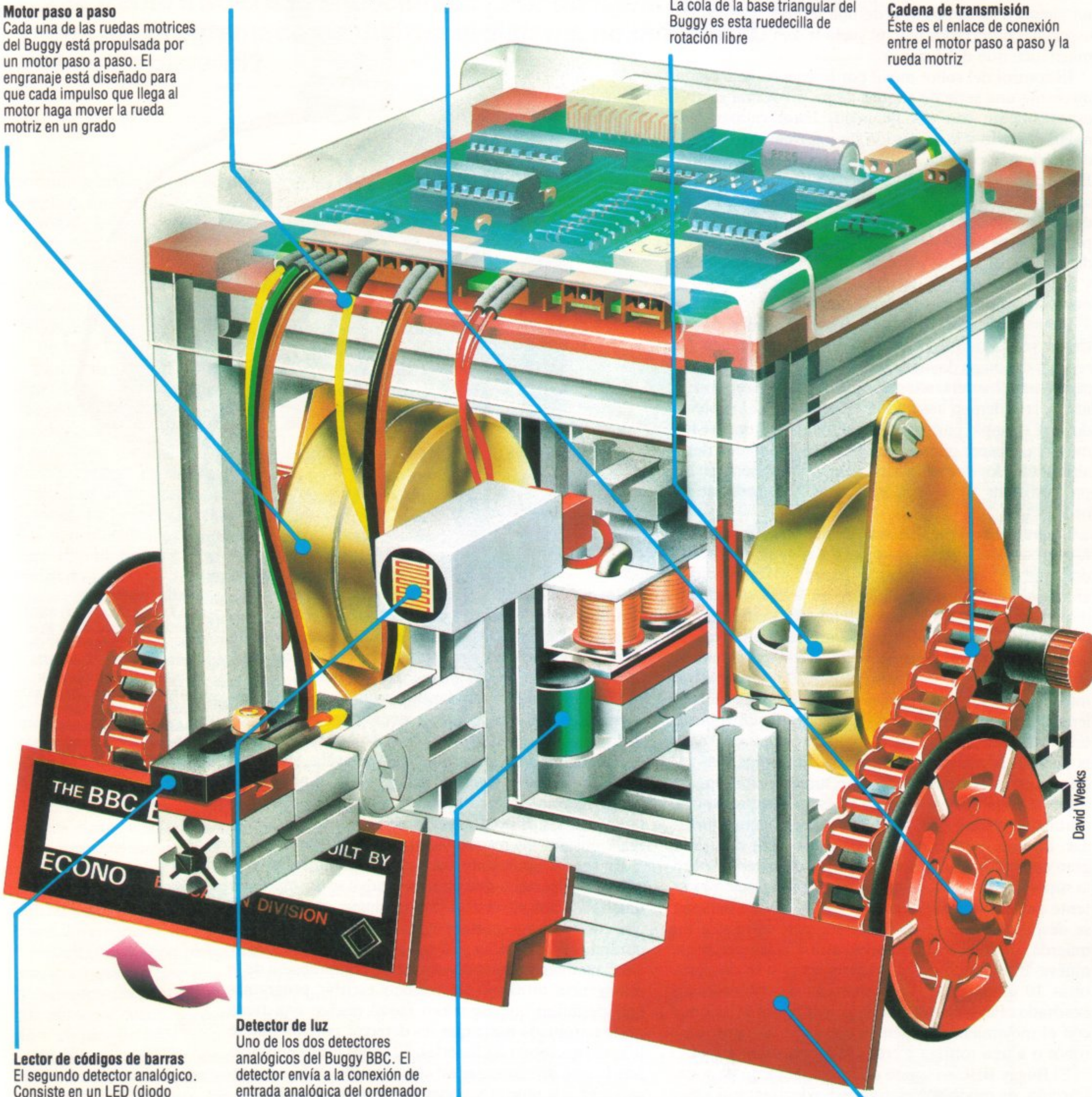
Otra aplicación para este tipo de detectores es el lector de códigos de barras. Los códigos relativos a la naturaleza de los productos almacenados en un depósito se podrían explorar del mismo modo que el robot busca el ítem correcto. El Buggy BBC viene con un software de demostración que emplea su lector de có-



En marcha

Esta "microtortuga" está en realidad a medio camino entre un robot móvil y una tortuga especializada, debido a que posee detectores de colisión y, en consecuencia, se le puede otorgar un nivel de "inteligencia". El Zeaker se puede desplazar hacia atrás, hacia adelante, hacia la izquierda y hacia la derecha, y también puede mover un lápiz hacia arriba y hacia abajo. Su control es muy sencillo gracias a una versión del lenguaje LOGO denominada "SNAIL LOGO", que permite dar directamente órdenes como FORWARD (hacia adelante) y BACKWARD (hacia atrás). A excepción del microordenador BBC, todos los otros ordenadores requerirán una unidad para interface. Esto representa un gasto extra, pero esta unidad posee su propia fuente de alimentación y, por tanto, no supone ninguna exigencia adicional para la fuente de alimentación eléctrica del ordenador

Chris Stevens



Rueda motriz
Las ruedas con cubiertas de goma situadas a cada lado del Buggy proporcionan el movimiento y evitan que el dispositivo patine, permitiendo que los desplazamientos sean precisos

Tablero de control
Todas las señales que llegan y salen del ordenador se procesan mediante esta ficha para interface antes de ser repetidas

Motor paso a paso
Cada una de las ruedas motrices del Buggy está propulsada por un motor paso a paso. El engranaje está diseñado para que cada impulso que llega al motor haga mover la rueda motriz en un grado

Ruedecilla trasera
La cola de la base triangular del Buggy es esta ruedecilla de rotación libre

Cadena de transmisión
Éste es el enlace de conexión entre el motor paso a paso y la rueda motriz

Lector de códigos de barras
El segundo detector analógico. Consiste en un LED (diodo emisor de luz) y un LDD (diodo detector de luz). La luz del LED se refleja desde la superficie por debajo del Buggy y es detectada por el LDD. La cantidad de luz reflejada depende del material y de su color, de manera que además de leer códigos de barras el detector también se puede emplear para seguir líneas trazadas en el suelo

Detector de luz
Uno de los dos detectores analógicos del Buggy BBC. El detector envía a la conexión de entrada analógica del ordenador una señal que es proporcional a la cantidad de luz detectada: cuanta más luz brilla en el Buggy, mayor es la señal

Portalápiz
El Buggy BBC se puede equipar con un portalápiz opcional entre las ruedas para que pueda trazar líneas por las superficies por donde pasa. El movimiento hacia arriba y hacia abajo del portalápiz se controla mediante el ordenador

Detectores frontales de colisión
Cada uno de los lados del parachoques delantero dividido está equipado con un microinterruptor que le proporciona al ordenador una señal de encendido (colisión) o apagado (no colisión)

digos de barras para ejecutar música; los principios son los mismos.

Otras clases de señales analógicas que le podrían resultar interesantes de seguir al robot móvil son luz, sonido y campos magnéticos. Estos campos se utilizan con frecuencia para las aplicaciones industriales en las cuales un robot ha de seguir un camino fijo a través de un almacén de depósito o de una fábrica. Los cables especiales enterrados bajo el suelo trazan el "camino" magnético que debe seguir.

El control del robot móvil por lo general se maneja mediante una serie de rutinas de un programa escrito especialmente con esa finalidad. Estas rutinas manejan el envío y la recepción de información a través del cable que conecta el dispositivo con el ordenador. En el caso del Buggy BBC, para controlar los motores se utilizan cuatro bits. Desde el Buggy los datos se envían a través del mismo cable. Las salidas analógicas provenientes tanto del sensor de luz como del lector de códigos de barras van hacia la conexión analógica, y los dos detectores de colisión están conectados a otras dos líneas de entrada al ordenador.

Tanto una conexión de I/O como la conexión dispositivos se pueden examinar mirando determinada dirección del mapa de memoria del ordenador. Generalmente para leer el contenido normal de la dirección se utiliza la orden en BASIC PEEK. Para alterar el contenido, por ejemplo para cambiar el sentido de uno de los motores y hacer girar el robot, el programador debe alterar el valor del bit adecuado en la dirección. En un programa en BASIC esto se conseguirá mediante la orden POKE.

La información analógica se puede examinar de forma bastante parecida, siempre y cuando el ordenador que se esté empleando posea un convertidor analógico-digital incorporado. Si no se dispone de tal dispositivo, se le ha de agregar al robot una unidad para interface que transforme las señales analógicas en información digital antes de reenviarlas al ordenador.

Una tortuga es, en realidad, una forma de robot móvil concebida para utilizarla en conjunción con el lenguaje LOGO, si bien la distinción entre un robot y una tortuga se está volviendo bastante confusa (véase la p. 164). Muchos de los modelos de tortuga más recientes están equipados con detectores de colisión, así como muchos robots móviles se están equipando con portalápices para que puedan funcionar como tortugas. La idea de que pueda ser posible crear dibujos a gran escala haciendo mover un lápiz con ruedas sobre la superficie surge de la enseñanza de la relación existente entre distancia, ángulo y forma. Si una persona se desplaza 10 unidades hacia adelante, gira a la izquierda, se mueve 10 unidades, gira a la izquierda, se mueve 10 unidades, gira a la izquierda y se desplaza otras 10 unidades, habrá caminado describiendo un cuadrado. Para ilustrar estas formas y sus relaciones con el movimiento, podemos acoplarle un lápiz a un robot o a una tortuga y crear la forma sobre papel.

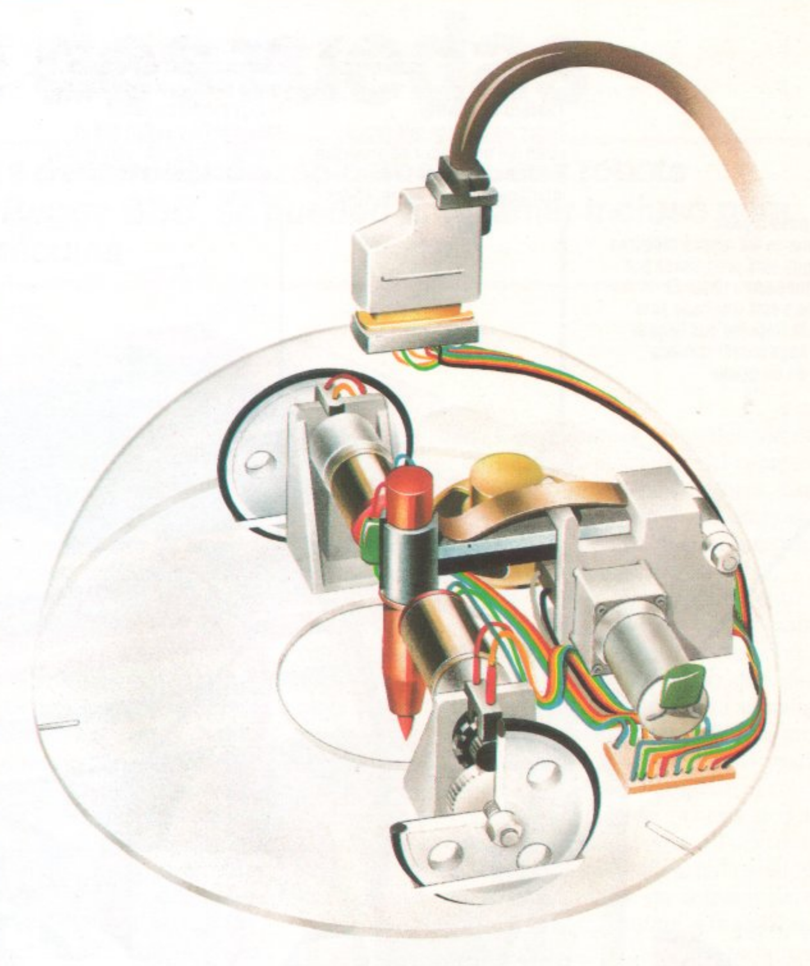
El Buggy BBC se vende en forma de juego de construcción, de modo que el usuario ha de montarlo antes de que pueda explorar el mundo de la robótica. El Buggy está basado en un sistema disponible a nivel comercial, el Fischer Technik, por lo cual ampliarlo y mejorarlo es muy sencillo.

Montar un dispositivo como el Buggy es, en sí mismo, una actividad educativa. Se puede aprender mucho examinando la forma en que se acoplan entre sí los diversos componentes. No obstante, el verdadero aprendizaje empieza cuando el usuario intenta asumir el control de su nuevo "juguete". Aunque muchos de los robots móviles disponibles en el comercio vienen con un software de control, resulta mucho más divertido escribir uno mismo los programas. Para ello se requiere enfocar la programación desde una perspectiva completamente nueva: la del control.

Mientras el robot está funcionando, el programa supervisa sus sensores constantemente para ver si ha hallado una línea en el suelo, si ha detectado una intensa fuente de luz, si ha chocado contra una silla, etc. En el instante en que el detector señala algo, el ordenador ha de reaccionar para proteger al robot de un daño potencial. Estos programas se denominan "en tiempo real" porque sus respuestas han de ser inmediatas.

En teoría existe muy poca diferencia entre un programa que permita que un robot móvil se desplace por una habitación sin chocar contra ningún objeto, y uno que controle una central eléctrica. Las técnicas que se aprenden jugando con dispositivos como el Buggy BBC también pueden desarrollar la comprensión de la inteligencia artificial. Se pueden escribir programas que permitan que un robot móvil realice una tarea predeterminada hasta que un detector perciba que se le están agotando las baterías. El robot busca entonces una fuente de alimentación eléctrica adecuada para recargarse a sí mismo y poder seguir funcionando.

La próxima generación de robots móviles ofrecerá configuraciones aún más notables. Probablemente estarán equipados con brazos en forma de garfios para que puedan coger y transportar cargas pequeñas. Asimismo, quizá los sensores luminosos se reemplacen por cámaras transistorizadas en miniatura gracias a las cuales el robot pueda "ver" a dónde se dirige. Las unidades para síntesis de voz le permitirán comunicarse con su operador, y el reconocimiento de voz abrirá otro canal de control sobre las acciones del robot.



David Weeks

Robot especializado
Una tortuga es como un robot especializado que dibuja en el suelo con un lápiz retráctil de punta de fieltro siguiendo las instrucciones del ordenador. Las tortugas normalmente se asocian con el lenguaje educativo LOGO (véase p. 164), si bien se pueden accionar por BASIC

"Dulces dieciséis"

Los ordenadores trabajan en notación binaria, dado que pueden hacer frente a sólo dos situaciones. ¿Por qué, entonces, algunas veces los programadores utilizan el sistema hexadecimal, que es de base 16?

Resulta sencillo comprender por qué los ordenadores emplean internamente el sistema binario: la representación de los números utilizando sólo ceros y unos se presta bien para las señales eléctricas encedido/apagado del ordenador. Asimismo, resulta sencillo comprender por qué la sociedad humana utiliza casi universalmente el sistema decimal: como tenemos diez dedos, este número se ha convertido en nuestro número de base natural.

Pero ¿para qué habría de servir el sistema hexadecimal (de base 16)? La razón es que convertir números hexadecimales en números binarios y viceversa es mucho más fácil que convertir decimales en binarios y binarios en decimales. Puesto que los ordenadores "comprenden" los números binarios, ¿por qué los informáticos no programan con dígitos binarios? Por dos razones. La primera de ellas es que los números binarios son mucho más largos que sus equivalentes de base 10 o de base 16. Por ejemplo, el número 356 en decimal es 0000000101100100 en binario. La segunda razón es que utilizando números que sólo consten de ceros y unos resulta mucho más fácil equivocarse.

El sistema hexadecimal utiliza 16 dígitos diferentes, incluyendo el cero, de manera que el número mayor que se puede representar en la columna de las "unidades" es el equivalente al decimal 15. Los matemáticos podrían haber inventado seis nuevos símbolos para los números entre 10 y 15, pero la convención exige emplear las seis primeras letras del alfabeto para los seis dígitos extra. Cuando contamos mediante la adición de unos, usamos los dígitos numéricos estándar junto con los dígitos alfabéticos hasta llegar a la F, el equivalente hexadecimal de 15. Después de ello, se nos terminan los símbolos y hemos de comenzar una nueva columna, la de los "dieciséis". Aquí también hemos de utilizar los símbolos disponibles hasta la F. Cuando tanto las columnas de los "dieciséis" como la de las "unidades" estén completas, sumar uno implicará que tengamos que agregar una tercera columna, la de los "doscientos cincuenta y seis". En el panel vemos algunos números decimales con sus equivalentes en binario y en hexadecimal.

En hexadecimal, el número 65 535 se representa FFFF, con lo cual resulta evidente una de las ventajas que ofrece este sistema. Un número cuya representación en decimal exige 5 dígitos y 16 en binario, en notación hexadecimal sólo requiere 4 dígitos. Aparte de esta compactibilidad, ofrece otra ventaja más importante. Cuatro dígitos binarios se pueden representar exactamente con un dígito hexadecimal. Esto hace que la conversión de binario a hexadecimal y viceversa sea relativamente directa.

Para transformar un número binario en su equivalente hexadecimal, divida el número binario en grupos de cuatro dígitos, empezando por la derecha, y convierta a hexadecimal un grupo cada vez. A continuación le ofrecemos algunos ejemplos:

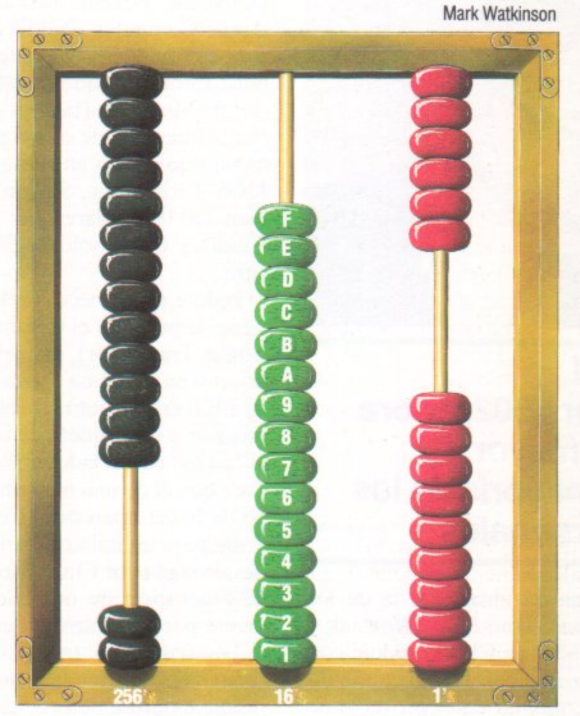
		1110	1001	binario
		=E	9	hex.
		=233		decimal
	1111	1000	1100	binario
	=F	8	C	hex.
	=3980			decimal
0111	1110	0010	1101	binario
=7	E	2	D	hex.
=32 301				decimal

¿Cuándo deseará realmente emplear la notación hexadecimal? A pesar de que los lenguajes de alto nivel, como el BASIC, no le exigen la utilización de hexadecimales (ni, para el caso, de binarios), sí lo exigen lenguajes de bajo nivel como el ensamblador y el código de lenguaje máquina.

Los números escritos en hexadecimal se suelen escribir seguidos de una H para diferenciarlos de los números decimales. Por lo tanto, 256 (en decimal) se escribiría 100H y se leería "uno cero cero hex.".

En esta obra utilizaremos muy poco los números hexadecimales. No obstante, puede que el usuario se encuentre con este término en los apéndices del manual de instrucciones del ordenador. Por eso conviene saber calcular su equivalente en decimal o en binario.

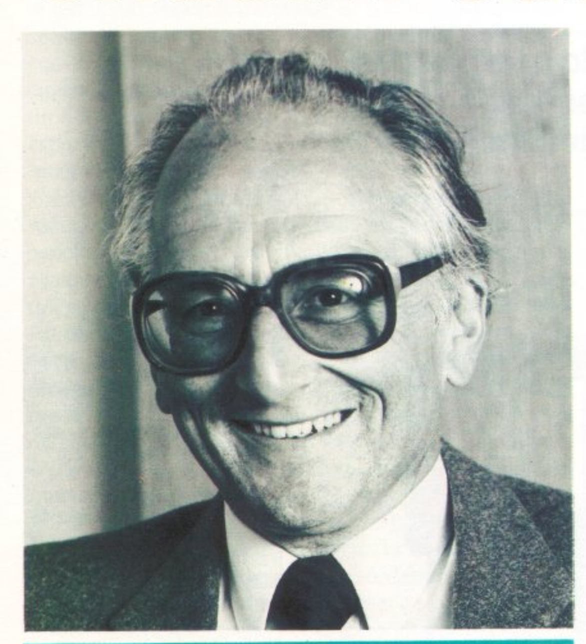
Mark Watkinson



El ábaco hexadecimal
Un ábaco construido para el sistema hexadecimal tendría 15 bolitas en cada varilla. En la varilla situada a la derecha, las bolitas se pueden hacer bajar de una en una hasta llegar a la última, la F. El hexadecimal es un sistema numérico de base 16 y para los números mayores que 9 se utilizan las letras del alfabeto. Como muestra la varilla central, A corresponde al decimal 10, B al 11, C al 12, D al 13, E al 14 y F al 15. Después del "número" F se nos terminan los dígitos y hemos de remitirnos a la columna siguiente, la de los "dieciséis". En la ilustración, la columna de los "1" nos muestra una cuenta de 9 bolitas. La columna de los "16" muestra una cuenta de F y la columna de los "256" una cuenta de 2. Esto se lee "2 F 9 hex.". Es el equivalente al número decimal 761
(256 × 2 + 16 × 15 + 1 × 9)

Binario	Decimal	Hex.
00000001	1	1
00000010	2	2
00000011	3	3
00000100	4	4
00000101	5	5
00000110	6	6
00000111	7	7
00001000	8	8
00001001	9	9
00001010	10	A
00001011	11	B
00001100	12	C
00001101	13	D
00001110	14	E
00001111	15	F
00010000	16	10
00010001	17	11
00010010	18	12
00010011	19	13
00010100	20	14
00010101	21	15

Chuck Peddle



Es el creador del microprocesador 6502, ahora fabricado al por mayor y utilizado por la mayoría de los ordenadores personales

Perteneciente a una generación anterior a la de los "jóvenes genios" empresarios como Steve Wozniak y Steve Jobs, el primer contacto de Chuck Peddle con los microprocesadores se produjo en 1973, cuando ingresó en la Motorola para trabajar en el proyecto de diseño del microprocesador 6800.

Puesto que se trataba del primer microprocesador que salía al mercado británico, Motorola pudo asignarle al 6800 un precio elevado: 2000 libras (unas 45 000 pesetas). Peddle consideró que el producto había sido sobrevalorado excesivamente y dejó la Motorola para integrarse a MOS Technology.

Se unió a esta relativamente pequeña compañía para trabajar en el proyecto de diseño de otro microprocesador, que se convertiría en el 6502 MPU, sin ninguna duda el microprocesador más exitoso de la primera década de la microinformática. No obstante, en aquel momento nadie comprendió que el producto en el cual estaban trabajando estaba llamado a convertirse en el baluarte de toda una industria y que contribuiría en gran medida al estallido de una revolución social como no se había visto desde hace dos siglos.

Una de las personas que comprendieron la trascendencia del microprocesador en general, y el potencial del MOS Technology 6502 en particular, fue Jack Tramiel, presidente de la Commodore. Hasta entonces, Commodore Business Machines se había dedicado a una gama de productos para oficina y calculadoras de bolsillo, con una aceptación discreta.

La Commodore era el principal cliente de MOS Technology, a la que compraba regularmente grandes cantidades de chips especializados para calculadoras de cuatro funciones. Tramiel, a pesar de las dificultades que él mismo tenía para mantener a flote la Commodore, tuvo suficiente fe en el 6502 como para conseguir de cualquier parte el capital necesario para adquirir MOS Technology. Simultáneamente con esta operación, contrató los servicios de Charles Peddle, que era a la sazón ingeniero de desarrollo de microprocesadores.

Ya en aquel entonces Peddle había comprendido que el producto que había desarrollado, fruto de sus propias ideas, podía tal vez marcar un hito: el ordenador personal. Era la misma idea que, de forma independiente, estaban alentando Wozniak y Jobs en la Apple Computer. Peddle estaba tan preocupado con el hecho de que la nueva tecnología se utilizara de manera apropiada, que se asoció con Bill Gates, fundador de Microsoft (famosa por su intérprete de BASIC), con la intención de comprar Apple, que, casualmente, había sido puesta en venta en el mismo momento que MOS Technology. Sin embargo, Wozniak y Jobs pedían 150 000 dólares por la empresa, y la oferta de Peddle y Gates sólo llegaba a los dos tercios de esa cifra.

Peddle permaneció en la Commodore y asumió la tarea de producir el Commodore PET (Personal Electronics Transactor), que apareció en 1977 en el mercado, casi en la misma época en que se lanzó el Apple II. El PET era distinto, en el sentido de que llevaba un monitor y un paquete de cassettes incorporado, y que el "tacto" del teclado se parecía más al de una calculadora que al de una máquina de escribir. Al poco tiempo de haber aparecido, Commodore tenía pedidos en firme para un millar de unidades, y, gracias a la labor desarrollada por Chuck Peddle, había nacido la primera generación de ordenadores diseñados específicamente para utilizarse en el hogar.

Transcurrieron tres años hasta que Peddle pudo concretar su segunda gran ambición: la de dirigir su propia empresa de ordenadores. Con Chris Fish, uno de los cerebros financieros que hicieron factible el súbito crecimiento de la Commodore, se asoció a Victor United, subsidiaria de la gigantesca Walter Kidde Corporation, y puso en marcha Sirius Systems Technology.

El trabajo de desarrollo en el campo de la industria de ordenadores personales se concentró de manera preferente en los chips de 16 bits, como el 8088 de Intel. Resultó que IBM también estaba trabajando en un ordenador personal para escritorio basado en el mismo chip, pero casualmente Sirius consiguió presentar el fruto de su trabajo algunas semanas antes. La máquina tuvo una amplia aceptación y pronto se afianzó en el mercado, siendo el primer microordenador barato y fabricado al por mayor que ofrecía las ventajas de la nueva generación de microprocesadores de 16 bits.

El Sirius 1 era relativamente barato y fácil de usar. Con su teclado separable, sus gráficos de alta resolución y su pantalla antideslumbrante, estableció nuevos niveles para los microsistemas de oficina. Los usuarios descubrieron la inmensa utilidad de las muy mejoradas velocidad y capacidad de direccionamiento del microprocesador de 16 bits.

En conjunto, Chuck Peddle ha recorrido un largo camino con el objetivo de hacer realidad su gran ambición: poner al alcance de todos el potencial del ordenador. Y en el proceso ha abierto nuevos caminos para que otros continúen su trabajo.



Cortesía de Sirius Computing

El primer ordenador personal
Después de desarrollar el microprocesador 6502, Chuck Peddle concentró sus esfuerzos en diseñar lo que se denominó "ordenador personal", una máquina autoconcentrada, que se podía enchufar a un tomacorrientes y utilizar de inmediato para cualquier finalidad que le interesara al usuario. El fruto de su trabajo fue el Commodore PET, que apareció casi simultáneamente que el Apple II de Steve Wozniak y llevaba pantalla, grabadora de cassette e intérprete de BASIC Microsoft incorporados. Aunque desde entonces ha sido rediseñado varias veces, el PET goza aún de gran popularidad. Una de las características más atractivas de la máquina original era su diseño. En la actualidad el 6502 es el microprocesador más utilizado para ordenadores personales

DONDE CONSEGUIR TU sinclair

ALAVA
COMPONENTES ELECTRONICOS GAZTEIZ
Domingo Beltran, 58 (Vitoria)
DEL CAZ
Avda. Gazteiz, 58 (Vitoria)
VALBUENA
Virgen Blanca, 1 (Vitoria)

ALBACETE
ELECTRO MIGUEL
Tesifonte Gallego, 27
TECON
Maria Marin, 13

ALICANTE
ASEMCA (Villena)
Avda. de la Constitución, 54 (Villena)
CONSULTING DESARROLLO INFORMATICO
Pais Valencia, 54 (Alcoy)
COMPONENTES ELECTRONICOS LASER
Jaime M.ª Buch, 7
ELECTRODATA LEVANTE
San Vicente, 28
ELECTRONICA AITANA
Limones, s/n. Edificio Urgull (Benidorm)
ELECTRONICA OHMIO
Avda. El Hamed, 1
LIBRERIA LLORENS
Alameda, 50 (Alcoy)

AVILA
FELIX ALONSO
San Segundo, 15

BADAJOZ
MECANIZACION EXTREMEÑA
Vicente Barantes, 18
SONYTEL
Villanueva, 16

BARCELONA
ARTO
C/ Angli, 43
BERENGUERAS
C/ Diputación, 219
CATALANA D'ORDINADORS
C/ Trafalgar, 70
CECSA
C/ Mallorca, 367
COMPUTERLAND
C/ Infanta Carlota, 89
COMPUTERLAND
Trav. de Dalt, 4
COPIADUX
C/ Dos de Mayo, 234
D. P. 2000
C/ Sabino de Arana, 22-24
DIOTRONIC
C/ Conde Borrell, 108
EL CORTE INGLES
Avda. Diagonal, 617-619
EL CORTE INGLES
Pza. Cataluña, 14
ELECTRONICA H. S.
C/ S. José Oriol, 9
ELECTRONICA SAUQUET
C/ Guilleries, 10
ELEKTROCOMPUTER
Vía Augusta, 120
EXPOCOM
C/ Villarroel, 68
GUIBERNAU
C/ Sepulveda, 104
INSTA-DATA
P.º S. Juan, 115
MAGIAL
C/ Sicilia, 253
MANUEL SANCHEZ
Pza. Major, 40 (Vic)
MILLIWATTS
C/ Melendez, 55 (Mataró)
ONDA RADIO
Gran Via, 581
RADIO ARGANY
C/ Borrell, 45
RADIO SONDA
Avda. Abad Margat, 77 (Tarrasa)
RAMEL ELECTRONICA
Cr. de Vic, 3 (Manresa)
REDISA GESTION
Avda. Sarriá, 52-54
RIFE ELECTRONICA
C/ Aribau, 80, 5.º, 1.ª
SERVICIOS ELECTRONICOS VALLES
Pza. del Gas, 7 (Sabadell)
SISTEMA
C/ Balmes, 434
S. E. SOLE
C/ Muntaner, 10
SUMINISTROS VALLPARADIS
C/ Dr. Ferrén, 172 (Tarrasa)
TECNOHIFI, S. A.
C/ La Rambleta, 19
VIDEOCOMPUT
P.º Pep Ventura, 9, Bl. C, Bjos. Bis (Vic)

BURGOS
COMELECTRIC
Calzada, 7
ELECTROSON
Conde don Sancho, 6

CACERES
ECO CACERES
Diego Maria Crehuet, 10-12

CADIZ
ALMACENES MARISOL
Camoens, 11 (Ceuta)
INFORSA
Avda. Fuerzas Armadas, 1 (Algeciras)
ELECTRONICA VALMAR
Ciudad de Santander, 8
M. R. CONSULTORES
Multi Centro Merca 80 (Jerez de la Frontera)
PEDRO VAREA
Porvera, 36 (Jerez de la Frontera)
LEO COMPUTER
Garcia Escamez, 3
SONYTEL
Queipo de LLano, 17
SONYTEL
José Luis Diez, 7
T L C Y AUTOMATICA
Dr. Herrera Quevedo, 2

CASTELLON
NOÙ DESPACH'S
Rey D. Jaime, 74

CIUDAD REAL
COMERCIAL R. P.
Travesera de Coso, 2 (Valdepeñas)
ECO CIUDAD REAL
Calatrava, 8

CORDOBA
ANDALUZA DE ELECTRONICA
Felipe II, 15
CONTROL
Conde de Torres Cabrera, 9
ELECTRONICA PADILLA
Sevilla, 9
MORM
Plaza Colón, 13
SONYTEL
Arte, 3
Avda. de los Mozárabes, 7

CUENCA
SONYTEL
Dalmacio Garcia Izcara, 4

GERONA
AUDIFILM
C/ Albareda, 15
CENTRE DE CALCUL DE CATALUNYA
C/ Barcelona, 35
S. E. SOLE
C/ Sta. Eugenia, 59

GRANADA
INFORMATICA Y ELECTRONICA
Melchor Almagro, 8
SONYTEL
Manuel de Falla, 3
TECNIGAR
Ancha de Gracia, 11

GRANOLLERS
COMERCIAL CLAPERA
C/ Maria Maspons, 4

GUIPUZCOA
ANGEL IGLESIAS
Sancho el Sabio, 7-9
BHP NORTE
Ramón M.ª Lili, 9
ELECTROBON
Reina Regente, 4

HUELVA
SONYTEL
Ruiz de Alda, 3

HUESCA
ELECTRONICA BARREU
M.ª Auxiliadora, 1

IBIZA
IBITEC
C/ Aragón, 76

JAEN
CARMELO MILLA
Coca de la Piñera, 3
MARA ILUMINACION
Avda. Linares, 13 (Ubeda)
MICROJISA
Garcia Rebull, 8
SONYTEL
José Luis Diez, 7
SONYTEL
Pasaje del Generalisimo, 3 (Linares)

LA CORUÑA
DAVIÑA
República de El Salvador, 29 (Santiago)
PHOTOCOPY
Teresa Herrera, 9
SONYTEL
Avda. de Arteijo, 4
SONYTEL
Tierra, 37

LAS PALMAS DE GRAN CANARIA
COMPUTERLAND
Carvajal, 4
CHANRAI
Triana, 3
EL CORTE INGLES
José Mesa y Lopez, 18

LEON
ELECTROSON
Avda. de la Facultad, 15
MICRO BIERZO
Carlos I, 2 (Ponferrada)
RADIO RACE
Modesto Lafuente, 3

LERIDA
SELEC
C/ Ferrer y Busquet, 14 (Mollerusa)
SEMIC
C/ Pi y Margall, 47

LUGO
ELECTROSON
Concepción Arenal, 38
SONYTEL
Primo de Rivera, 30

MADRID
ALFAMICRO
Augusto Figueroa, 16
BELLTON'S
Torpedero Tucuman, 8
CHIPS-TIPS
Pto. Rico, 21
CMP
Pto. Santa Maria, 128
COMPUTERLAND
Castelló, 89
COSESA
Barquillo, 25
DINSA
Gaztambide, 4
DISTRIBUIDORA MADRILEÑA
Todos sus centros
ELECTROSON
Duque de Sexto, 15 (y otros centros)
INVESMICROSTORE
Génova, 7
J.P. MICROCOMPUT
Montesa, 44
EL CORTE INGLES
Todos sus centros
ELECTRONICA SANDOVAL
Sandoval, 4
PENTA
Dr. Cortezo, 12
RADIO CINEMA
Antonio Acuña, 3
RADIO QUER
Todos sus centros
SONYTEL
Clara del Rey, 24 (y todos sus centros)
SONICAR
Vallehermoso, 19
VIDEOMUSICA
Orense, 28

MALAGA
EL CORTE INGLES
Prolongación Alameda, s/n.
INGESCON
Edificio Galaxia
SONYTEL
Salitre, 13

MELILLA
OFI-TRONIC
Hermanos Cayuela, 11

MENORCA
ELECTRONICA MENORCA
C/ Miguel de Veri, 50 (Mahon)

MURCIA
COMPUTER LIFE
Alameda San Antón, 2 (Cartagena)
EL CORTE INGLES
Libertad, 1
ELECTRONICA COMERCIAL CRUZ
Rio Segura, 2
MICROIN
Gran Via, 8

NAVARRA
ENER
Paulino Caballero, 39
GABINETE TECNICO EMPRESARIAL
Juan de Labrit, 3
JOSE LUIS DE MIGUEL
Arrieta, 11 bis

OVIEDO
AUTECA
Valentin Masip, 25
EDIMAR
Cangas de Onis, 4-6 (Gijón)
ELECTRONICA RATO
Versalles, 45 (Aviles)
RADIO NORTE
Uria, 20
RESAM ELECTRONICA
San Agustin, 12 (Gijón)
RETELCO
Cabrales, 31 (Gijón)
SELECTRONIC
Fermin Canellas, 3

ORENSE
SONYTEL
Concejo, 11

PONTEVEDRA
EL CORTE INGLES
Gran Via, 25 (Vigo)
ELECTROSON
Santa Clara, 32
ELECTROSON
Venezuela, 32 (Vigo)
SONYTEL
Salvador Moreno, 27
SONYTEL
Gran Via, 52 (Vigo)
TEFASA COMERCIAL
San Salvador, 4 (Vigo)

PALMA DE MALLORCA
GILFT
Via Alemania, s/n
IAM
C/ Cecilio Metlo, 5
TRON INFORMATICA
C/ Juan Alcover, 54, 6.º C

LA RIOJA
YUS COMESSA
Cigueña, 15

SALAMANCA
DEL AMO
Arco, 5
PRODISTELE
España, 65

SANTANDER
LAINZ S. A.
Reina Victoria, 127
RADIO MARTINEZ
Dr. Jiménez Diaz, 13

SEGOVIA
ELECTRONICA TORIBIO
Obispo Quesada, 8

SEVILLA
A.D.P.
San Vicente, 3
EL CORTE INGLES
Duque de la Victoria, 10
SCI
Aceituno, 8
SONYTEL
Pages del Corro, 173
Adriano, 32

TARRAGONA
AIA
Rambla Nova, 45, 1.º
CIAL. INFORMATICA TARRAGONA
C/ Gasometro, 20
ELECTRONICA REUS
Avda. Prat de la Riba, 5 (Reus)
SEIA
Rambla Vella, 7 B
S.E. SOLE
C/ Cronista Sese, 3
T. V. HUGUET
Pza. Major, 14 (Montblac)
VIRGILI
C/ Dr. Gimbernat, 19 (Reus)

STA. CRUZ DE TENERIFE
COMPUTERLAND
Mendez Nuñez, 104 B
TRENT CANARIAS
Serrano, 41

VALENCIA
ADISA
San Vicente, 33 (Gandia)
CESPEDES
San Jacinto, 6
COMPUTERLAND
Marqués del Turia, 53
DIRAC
Blasco Ibáñez, 116
EL CORTE INGLES
Pintor Sorolla, 26
Meléndez Pidal, 15
PROMOCION INFORMATICA
Pintor Zariñena, 12

VALLADOLID
SONYTEL
León, 4

VIZCAYA
BILBOMICRO
Aureliano del Valle, 7
DATA SISTEMAS
Henao, 58
DISTRIBUIDORA COM
Gran Via, 19-21 y todos sus centros
EL CORTE INGLES
Gran Via, 9
ELECTROSON
Alameda de Urquijo, 71
San Vicente, 18 (Baracaldo)
GESCO INFORMATICA
Alameda de Recalde, 76
KEYTRON
Hurtado de Amezaga, 20

ZAMORA
MEZZASA
Victor Gallego, 17

ZARAGOZA
EL CORTE INGLES
Sagasta, 3
SONYTEL
Via Pignatelli, 29-31



DISTRIBUIDOR EXCLUSIVO:
INVESTRONICA

Central Comercial
TOMAS BRETON, 60
TELF. 468 03 00
TELEX 23399 IYCO E
MADRID

Delegación Cataluña
MUNTANER, 565
TELF. 212 68 00
BARCELONA